China-pub.com
下载

第四部分　Lingo基础

第12章　学习Lingo

Director的意义远远超过一个制作动画片或活动演示的工具。我们可以用它制作完整的应用程序、Web短程序和软件工具，制作游戏、教学软件和商用软件等。几乎所有可以用软件完成的事情都能用Director完成。

用Director编写功能强大的软件的关键角色是Lingo —— Director的编程语言。起初，Lingo只是一些用来控制动画的简单的剧本命令。但现在，它已经是完整的面向对象的编程语言，胜过了传统的编程语言如C或Pascal，也胜过了新的编程语言如Java。第13章“重要的Lingo句法”将继续讨论这个话题，并列出所有我们应该熟悉并了解的重要Lingo命令。

12.1　什么是Lingo

Lingo是一种编程语言，是通过给计算机发命令、提问题而与计算机对话的一种方法。在Director中，与我们对话的是Director环境。

Lingo是一种类似英语的编程语言。我们可以大声读出Lingo代码行，并根据其英语单词的含义理解它。Director中所有的命令、函数及其他关键词都是英语的单词、词组或缩写。这使得Lingo比其他编程语言更容易学习。

Lingo是一种演员类型。也就是说当我们创建了控制影片的Lingo代码后，它将存储在称为剧本(Script)的演员中。剧本就是包含一些文本的演员，这些文本都是有效的Lingo代码。在Director的演员表中，它们与位图、声音和图形等共同存在。有时，它们也被放在剪辑室内。

Lingo的功能是强大的。在所有的Director产品中，只有少数产品不需要使用Lingo。用其他软件能够完成的所有任务几乎都能用Director和Lingo实现。尽管在有些情况下，对于特定的任务来说，Lingo不是最好的工具；但在其他情况下，与传统方法相比，用Lingo完成某软件的编程将更容易。

学习Lingo是完整地使用Director的关键所在。如果你不喜欢枯燥的PowerPoint风格的演示或直来直去的简单动画，就该学习Lingo，以便更好地使用Director。

所幸Lingo很容易学。该语言里有几百个关键词，但只要学习几个就可以入门了。我们可以从Lingo消息(Message)窗口开始。

12.2　消息窗口初步

尽管Lingo剧本通常存储在演员里，但我们也可以在消息窗口中执行短的、单行的程序。选择Window｜Message，或按组合键Command+M (Mac)或Ctrl+M (Windows)，可以打开消息窗口。

消息窗口很普通，其中有一大片区域是用来录入和观察文本的。在消息窗口打开的情况

China-pub.com 下载

下打字，字符就会出现在窗口内。但它不仅仅能对打字有所反应，它实际上是一个 Lingo翻译器，能够把Lingo命令与剧本(我们不久就将学到如何写剧本)等同对待。

现在向消息窗口中键入 put 42。当你按回车后，消息窗口将翻译你刚才键入的命令，并在下一行返回 --42。图12-1就是此时的消息窗口。

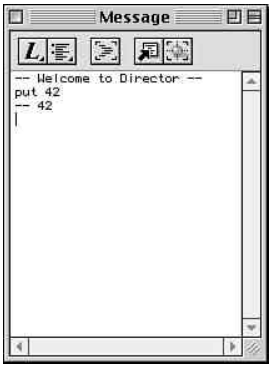

图12-1 消息窗口允许我们键入单行Lingo代码，并观察结果

在消息窗口中，双连字符“ -- ”出现在Director返回给我们的代码行之前。在后面，我们将要学习到如何使用双连字符来注释代码。

现在我们已经知道了第一个 Lingo命令：put。put命令用来把文本放在消息窗口中。在后面我们还将了解到它的其他用途。

我们要求Director把数字42放(put)在消息窗口中，于是它就那样做了。但它所做的并不是简单地把一个数字原样返回。事实上它理解 42是什么。要验证这一点，可以试试下面的代码：

```
put 42+1
--43
```

现在我们看到Director会做加法运算。它并不单单地把“ 42+1 ”返回来，而是明白这些是数字，而且加号意味着应该把这些数字相加。

除数学运算以外，还可以用Lingo做其他事。在消息窗口中试着输入：

```
beep
```

你可以听到计算机系统的缺省的警告声“ 嘀 ” (如果没有听到，可能是由于音量过小，或系统的警告声功能已被关闭)。可以注意到，消息窗口并没有在下一行返回任何文本，这是因为我们并没有要求它这样做。命令 beep只是让系统发出警告声。它不像 put命令那样会在消息窗口返回一些内容。

消息窗口对我们初次接触 Lingo很有益，而且在我们学习新的 Lingo命令时，它仍旧十分有用。即使是Lingo编程专家也经常使用消息窗口进行 Lingo编程。

参见第13章“ 重要的Lingo句法 ”里的13.1.2节“ 整数和浮点数 ”，可以获得更多有关在Lingo中使用数字的信息。

12.3 理解剧本的类型

消息窗口允许我们每次键入或翻译一行 Lingo代码，但我们需要同时把多行 Lingo代码串起来使用，才成其为程序。多行 Lingo代码被存储在称为剧本(Script)的演员里。有三种不同类型的剧本演员：影片剧本(Movie Script)、行为剧本(Behavior Script)和父代剧本(Parent Script)。此外，还可以把剧本赋予位图等其他演员，这类剧本通常称为演员表剧本 (Cast Script)。

所有这些剧本类型的区别不在于它们的外观或特性，而在于它们何时起作用。

例如，影片剧本是在整部影片中都起作用的剧本。如果影片剧本这样写：每次点击鼠标时，都发出一声系统的警告声，那么无论影片演到何处，只要鼠标被点击，就发出警告声。

行为剧本在概念上与我们在第 2章“ 用Director制作演示 ”中所遇到的行为相似。在它被

China-pub.com
下载

放到角色上或放在帧剧本通道内之前，它不起任何作用。当它被放到角色上时，剧本内的Lingo命令只有在牵涉到角色时才起作用。如果有一个行为，其内容是当点击鼠标时计算机就发出警告声，而且该行为被赋予某个角色后，只有当用户用鼠标点击那个角色时，计算机才会发出警告声。由于这个原因，行为剧本有时也称为角色剧本或剪辑室剧本。它们只对它们所附属的剪辑室里的角色起作用。

行为剧本还可以被赋予剪辑室中的帧剧本通道。在这种情况下，它们的作用与影片剧本相似，但仅对它们所附属的帧起作用。有着这种用途的行为剧本有时也被称作剪辑室剧本。

父代剧本是一种更高级的剧本。在我们使用一些面向对象的 Lingo命令告诉它们如何以及何时被使用之前，它们几乎不起任何作用。

相比之下，演员表剧本使用起来非常容易。创建演员表剧本的方法是：选择一个演员，如一幅位图，再点击演员表窗口里的剧本按钮。其结果是打开了剧本窗口，允许我们为那个演员添加剧本。

演员表剧本只对那一个演员起作用。如果为一个演员写的剧本要求每当用户点击鼠标时，计算机系统就发出警告声，那么只有当该演员在舞台上时，用户的操作才会影响它。如果那个演员在剪辑室里被使用了不止一次，在它每次被使用时该剧本都将起作用。

在现代Lingo编程中已经不使用演员表剧本了。用行为就可以完成同样的任务，而且更具灵活性。在本书中也很少用到演员表剧本。

要创建剧本，先在演员表窗口内选择一个空的位置，再按组合键 Command+0(零) (Mac)或Ctrl+0 (Windows)。剧本窗口将出现。这个窗口非常简单，但在顶部的几个按钮里隐藏着几个窍门。当我们编辑另一类演员的剧本时，剧本窗口也会出现。图 12-2是剧本窗口。

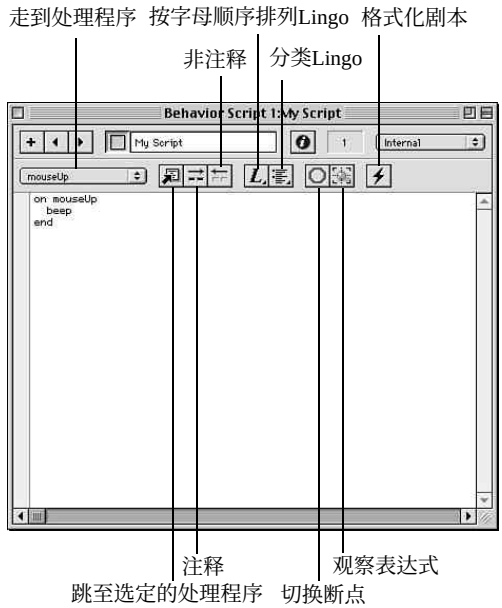

图12-2 剧本窗口允许我们编辑影片剧本、行为和父代剧本

在剧本窗口的最顶部有典型的 Director演员按钮，用它能够在演员表中向前或向后移位，允许我们添加新演员和切换演员表库。其余的按钮与更高级的 Lingo功能(如处理器和调试等)有关，等到介绍那些功能时再介绍有关的按钮。

China-pub.com 下载

按字母排列Lingo和分类Lingo这两个按钮允许我们在剧本中寻找或向剧本中自动插入Lingo关键字。当我们忘记了某个命令的全名，而只记得它的开头字母或它所属的类别时，使用该功能就非常方便。它还可以随时帮助我们准确地记忆Lingo命令的句法。

提示 在使用按字母排列Lingo和分类Lingo按钮的弹出菜单的同时，按下Option(Mac)或Alt(Windows)键，可以快速地调出相应命令的帮助信息。

剧本演员有一个Script Cast Member Properties(剧本演员属性)对话框，如图12-3所示。剧本演员只有两个属性，即演员的名称和它的剧本类型。三个可选项当然就是：Movie、Behavior和Parent。

图12-3 在Script Cast Member Properties对话框中，可以改变剧本的类型

12.4 使用消息和处理程序

在图12-3的剧本窗口中，有一个很小、很简单的剧本，其内容是：

```
on mouseUp
  beep
end
```

你可能还记得mouseUp消息，在第11章“高级技巧”中曾经用过它。

消息是由某个事件发出的一个信号。例如，当用户按下鼠标键，后来又释放它时，就发出了一个mouseUp消息。与此相似，起初按下鼠标键时，也发出了一个mouseDown消息。

这类消息会发给与此动作相关的任何Director对象。例如，mouseUp消息将发给被点击的角色。如果用户点击的位图是舞台上的角色3，该角色将得到一个mouseUp消息。如果那个角色附带有一个行为，Director将检查那个行为是否处理mouseUp消息。

行为或剧本是通过定义处理程序(handler)来处理这些消息的。处理程序以on开头。在前面那个例子里，处理程序的第一行是on mouseUp。这表明该处理程序是用来处理发送给那个角色的任何mouseUp消息的。

处理程序的内容——在此处是单个命令beep——是Lingo命令。如果处理程序被消息触发，就执行该命令。这一系列的命令由end结尾，即表示“这里是处理程序的结尾”。

注释 很多程序员喜欢把处理程序的名字放在end后面。这样，图12-3里的最后一行程序就变成end mouseUp。Lingo并不需要这样，但有些程序员仍旧这样做。

在剧本里可以放置多个处理程序。例如，在某个行为里，可以同时存在有on mouseUp和on mouseDown两个处理器。它们是受不同消息触发的，因此不会相互干扰。

参见第14章“创建行为”里的14.2“创建简单行为”，可以得到更多有关创建行为的信

China-pub.com
下载

息。

12.4.1 消息的类型

除了mouseUp和mouseDown外，还有其他许多类型的消息。它们会响应 Director中发生的事件。下面列出了最常用的事件和消息：

■ mouseDown —— 用户点击鼠标键。在Windows中，这对应于鼠标的左键。

■ mouseUp —— 用户释放鼠标键。每一个 mouseUp消息前面都有一个 mouseDown消息，但它们不一定是发给同一个角色的。

■ mouseEnter —— 光标进入某个角色的区域。

■ mouseLeave —— 光标离开某个角色的区域。

■ mouseWithin —— 只要光标在角色上，就连续发送该消息，即每帧发送一次。

■ mouseUpOutside —— 当用户点击某个角色，却又把光标移出，使之离开对象时所发送的消息。

■ beginSprite —— 在剪辑室内，当播放头首次遇到某角色时所出发的消息。

■ prepareFrame —— 当新的一帧刚开始、还没有显示在舞台上时所发出的消息。

■ enterFrame —— 当新的一帧刚开始、已经显示在舞台上时所发出的消息。

■ exitFrame —— 当影片播放完一帧，准备播放下一帧时所发出的消息。

■ prepareMovie —— 在显示影片的第一帧之前所发送的消息。

■ startMovie —— 在刚刚显示影片的第一帧后立即发送的消息。

■ stopMovie —— 当影片结束时所发出的消息。

■ idle —— 在影片从播放开始到播放结束的过程中持续不断地发出的消息。发出该消息的次数取决于节奏有多慢和计算机的速度有多快。

■ keyDown —— 当用户按下键盘上的某个键时所发出的消息。

■ keyUp —— 当用户释放键盘上的某个键时所发出的消息。

12.4.2 消息的接收顺序

除了要知道Director的事件能够发送的各种消息外，更重要的是理解这些消息的去向。例如，一个mouseUp消息可以由某个被点击的角色所附带的行为接收。不过，如果该角色的没有含有on museUp处理程序的行为，该消息将继续寻找能够接收它的处理程序。

该消息寻找的下一个地点是演员表。它查看在演员表剧本中是否有 on mouseUp处理程序。如果没有，它将查看帧剧本通道内的行为。如果还没有，最后它将在影片剧本演员中寻找 on mouseUp处理程序。

如果仍旧没有找到接收该消息的处理程序，该消息就没有用了。以下列出消息寻找处理程序的过程：

■ 正在演出的某个角色所附带的行为。如果有多个行为，将按照这些行为在行为监察窗内的排列顺序寻找。

■ 演员表剧本。

■ 帧剧本通道内的行为。

■ 任何影片剧本演员。

China-pub.com 下载

在有些情况下，由于明显的原因，上述的一步或多步过程将被放弃。例如， exitFrame消息与任何角色无关，因此Director直接把它送到帧剧本通道。 startMovie消息仅寻找影片剧本。

12.4.3 创建处理程序

前面列出了Director原有的事件及其发出的消息。实际上，我们可以创建和发送我们自己的消息，也可以创建接收这些消息的处理程序。

创建一个影片剧本，并在其中放置一个简单的处理程序：

```
on playBeep
  beep
end
```

要确认它是一个影片剧本，并关闭剪辑室窗口，然后打开消息窗口。

注释 Lingo有一个约定，即用几个单词来定义处理程序或变量，如“ my number ”或“ play beep ”。但由于处理程序的名称中不能带空格，为了使这些名称的可读性好，把第一个单词后面的每个单词的开头字母大写。于是“ my number ”变成了myNumber ”，“ play beep ”变成了“ playBeep ”。其实Lingo并不关心大小写问题，因此这样写只是为了好看(myNumber和mynumber对于Lingo来说是完全相同的)。

消息窗口的另一个功能是向影片发送消息。键入消息的名称即可完成这件事。如果一个影片剧本含有处理该消息的处理程序，该处理程序就会运行。否则，将出现一个错误信息，告诉我们未定义接收该信息的处理程序。

当我们在消息窗口内键入playBeep时，playBeep处理程序将运行，发出系统警告声。

12.5 使用变量

Lingo以及每一种编程语言中的另一个关键元素是变量。变量是存储数值的区域。

例如，可以把42存储在名为myNumber的变量里。要实现这个目标，只要用“ = ”号把这个值赋给变量。在消息窗口中试验：

```
myNumber=42
```

注释 在Director 7以前，我们需要用Lingo命令set来为变量赋值。Director不再需要这样了，但仍旧允许这类旧的句法存在。

在我们键入上面这行代码前， myNumber对于Director和Lingo来说都没有意义。这行代码的意思是告诉Director创建一个名为myNumber的变量，并把数字42存储在其中。现在，我们可以用put命令得到该数值，把它显示在消息窗口中。

```
put myNumber
--42
```

Director记得我们曾把数值 42存储在名为 myNumber的变量里，于是当我们让它把myNumber放在消息窗口内时，它从它的存储区域找到了这个名为 myNumber的变量，并把它的值放在了消息窗口。还可以用变量做更复杂的事情。键入以下内容：

```
myNumber = 42+1
put myNumber
-- 43
```

China-pub.com
下载

Director在放置变量的值之前，先进行算术运算。我们还可以用已被赋值的变量进行算术运算。

```
myNumber = 5
myOtherNumber = 3
put myNumber+myOtherNumber
-- 8
```

也可以改变已经存在的变量的值。

```
myNumber = 42
put myNumber
-- 42
myNumber = myNumber+1
put myNumber
-- 43
```

数字并不是变量所能存储的唯一一种内容，它们也可以存储字符。试输入以下内容：

```
myName = "Gary"
put myName
-- "Gary"
```

一系列的字符叫做字符串。它总是带有引号，实际上是 Lingo要求必须有引号。因此一个数字 —— 如42 —— 可以直接写成42，而字符串则必须带有引号。

一个字符串可以只有1个字符长甚至0个字符长(“ ”)。对字符串的长度值的最大值没有限制(只要计算机的内存允许)。

变量也可以用在处理程序中。例如：

```
on playWithVariables
  myNumber = 5
  myNumber = myNumber+4
  myNumber = myNumber-2
  put myNumber
end
```

如果把该处理程序放在一个剧本里，然后在消息窗口中键入 playWithVariables，我们将看到数字7被放在消息窗口内。

这样使用的变量叫做局部变量，即它只用于一个处理程序内部。它只在该处理程序被使用时存在，当该处理程序运行结束时，它就不存在了。若该处理程序再次被调用，这个变量将重新被创建。

如果我们创建了另一个处理程序，它也使用名为 myNumbr的变量，这个变量与刚才的那个是完全不同的两个变量。每一个处理程序就是一个属于它自己的小天地，其内部的局部变量只属于它自己，而不属于其他处理程序。

我们可以创建另一种变量，即全局变量，它可以由多个处理程序共享。下面举一个全局变量的例子。以下是一个影片剧本的完整内容，它包括 3个处理程序：

```
on initNumber
  global myNumber
  myNumber = 42
end

on addOneToNumber
  global myNumber
```

China-pub.com 下载

```
  myNumber = myNumber+1
end

on putNumberInMessageWindow
  global myNumber
  put myNumber
end
```

注释 可以不在每个处理程序内都用global命令声明全局变量，而在所有处理程序之外——如剧本的最前面几行——放置一个global命令。这为该剧本演员内的所有变量定义了一个全局变量。

由于这三个处理程序都用 global命令表示myNumber是一个全局变量，因此它们都可以使用这个变量。因此我们可以在消息窗口内这么做：

```
initNumber
putNumberInMessageWindow
-- 42
addOneToNumber
putNumberInMessageWindow
-- 43
```

上面那段代码演示了各个处理程序是如何引用同一个变量的。现在删去其中的 global行，看看会得到什么结果。

提示 可以在消息窗口中或在某个处理程序中用 clearGlobals命令删除所有全局变量。还可以在消息窗口中用showGlobals命令列出一个清单，其中含有当前的全部全局变量和它们的值。

12.6 编写Lingo程序

用Lingo制作Director影片有多种不同的方法。随着时间的推移，你将找到你自己最习惯的方法。

例如，有的程序员习惯于创建一个影片剧本演员，其中包含所有的影片处理程序。其他一些程序员则按某种分类方法把它们分在几个剧本演员里。有些程序员甚至为每个处理程序创建一个剧本演员。

如果你经常编程，会发现你的编程风格也会经常变化。有时没有严格的对与错之分。不过，一些基本的编程规则可以帮助你有一个好的开始。

12.6.1 Lingo程序员

Lingo程序员可分为两类。一类是有计算机学的背景知识的。他们大都了解C或Pascal语言，并在大学修过“数据结构”和“线性代数”等课程。

第二类程序员占大多数。他们是图形图像艺术家或多媒体作品的创作者，这些程序员从前可能使用过制作演示讲稿的工具，甚至用过 Director，但从未使用过编程语言。他们现在也打算学习Lingo了。

对于这两种类型的程序员来说，开始学习 Lingo时都会遇到困难。对于有编程经验的程序员来说，Lingo接管了一些他们原本已习以为常的繁琐的工作，但需要他们能够控制图形图像

China-pub.com
下载

元素和用户界面。对于图形图像艺术家来说， Lingo只是一行又一行文字，像拦路虎一样站在他们和他们的作品之间。

需要记住的重要的一点是：编程也是一种技术。像 Lingo这样的编程语言为程序员表达他们自己的思想提供了一方自由的天地。如果把一个任务同时交给两个程序员去完成，他们写出的程序很可能不同。每个程序都反应了它的作者的风格。

对于有编程经验的程序来说， Lingo为他们提供了另外一种环境，使他们能在其中进行创作。同样身为一个有经验的编程员，我敢说 Lingo能够让他们比从前更有创造力。

对于图形图像艺术家来说， Lingo就像一把新的画笔。很多我认识的艺术家都开始用Lingo编程。Lingo成为他们把创意变为艺术作品的一种新方法。

12.6.2 编程与解决问题

编程就是在解决问题。如果我们想制作一个 Director影片，其内容是一个配对游戏，就要把它想成是一个问题。而我们的最终目的就是解决这个问题。

同所有问题一样，这个问题的解决也需要许多步骤。我们要分析这个问题，把它分成许多部分，看看我们的已知和未知条件。然后，需要列出一个解决该问题的计划。

要解决一个编程问题，首先要明确地描述它。要问自己：“ 我究竟想要影片中发生什么样的故事？ ”如果回答：“ 想让一个角色活动起来 ”，这并没有明确地描述我们所面临的问题；如果回答：“ 我想把一个角色从舞台的左边移动到舞台的右边 ”，这只能算是稍微好一点的回答。真正的描述应该是这样的：“ 我想让角色以10fps的速度花5秒钟从坐标50 200处移动到550 200处。

在问题被描述出来后，就开始思考如何解决它了。假设我们的目的是以 10fps的速度在5秒钟内把一个角色移动 500像素。在10fps的速度下， 5秒钟共有50帧。因此要在1帧内把角色移动10像素。

只有清晰地描述了问题，才能开始思考如何解决它。

12.6.3 化解大的问题

用任何语言编程的关键之处都在于能够把问题分解。我们起初遇到的总是一个很大的命题。例如，“ 帮助儿童学习地理知识的测验 ”，这个问题包含的内容就很多。我敢打赌 Lingo里决没有quizKidsOnGeography命令。

因此我们把它分解成小问题。我们希望在每个问题中，都显示一个世界地图，其中有一个国家是突出显示的。然后，屏幕上出现三个可供选择的答案，让参加测验的孩子从中选出正确的那一个。因此现在可以暂时不考虑整个的程序，而只集中解决当前的问题。

但这个问题还是太大，不能一次解决。不过，它还可以被拆分。如何在地图中突出显示某一个国家？三个选择答案如何出现？如何保证三个选择答案中有一个是正确的？

把问题拆分后，得到的一个小问题之一可以是：用 Lingo程序实现从100个国家名称当中随机抽选一个。由于现在我们已经把问题分解到了这个程度，于是就可以开始编程了。该段程序的任务是得到一个处理程序，它随机抽选一个国家，并将其显示在消息窗口内。编出这段小程序，再对它进行测试，就解决了第一个小问题。

接着，用刚才的方法找出并解决其他一些小问题。不知不觉中，我们就得到了一个近乎完整的程序。

China-pub.com 下载

把大的问题拆分成小问题是编程工作中的一个重要环节。如果我们在 Director中编程的过程中被困住，那很可能就是因为没有把问题拆分得足够小。退一步，再分析所面临的问题，并决定在继续前进之前，如何把问题再拆成更小的问题。

12.6.4 创建剧本演员

如果我们要创建一个小的短程序或放映机，管理剧本的一个有效的方法设立一个影片剧本演员和几个行为演员。

也可以不使用任何影片剧本。如果行为写得好，就可以不需要影片剧本。

如果某部影片的大部分内容都是按钮，用一些行为剧本就可以处理这个问题。每个行为可以附属于一个按钮，告诉它在用户点击这个按钮时应该怎么做。

一部大型的Lingo作品将需要几个影片剧本。例如， onStartMovie处理程序中的一些项目就需要影片剧本。在这个处理程序里有一些影片刚开始播放时所需要执行的命令。

影片剧本内还可以容纳由多个行为所使用的处理程序。例如，如果我们想让多个行为都发出随机的声音，就可以让它们调用位于影片剧本内的 on playRandomSound处理程序。这样就避免了在许多行为内都放置功能相似的处理程序。

在演员表里，把影片剧本集中放置比较好。对于行为也应这样处理。当然也有例外，有时我们可以把行为放在它们经常要控制的位图演员的附近。

12.6.5 编写程序

Director 7有自动为剧本上色的功能。该功能可以为剧本中的不同类型的词上不同的颜色。其目的是让我们更容易地读程序。也可以在 Script Preferences(剧本参数设置)对话框里把该功能关闭。这时，又可以用文本监察窗为程序中的文本手工地上色和设定格式。

写程序时要记住的最重要的事情是添加注释。注释就是在程序中添加的单词、短语或句子，它们标注了程序代码的作用。注释可以自成一行，也可以接在代码行后面。用双连字符告诉Director，在运行Lingo时，所有在双连字符后面的文字都将被忽略。例如：

```
-- This handler output powers of 2 to the Message Window
on powersOfTwo
  n = 2 -- start with 2
  repeat with i = 1 to 100 -- output 100 numbers
    put n -- send to the Message window
    n = n*2 -- multiply by 2 to get the next number
  end repeat
end
```

比较一下不带注释的同一个处理程序：

```
on powersOfTwo
  n = 2
  repeat with i = 1 to 100
    put n
    n = n*2
  end repeat
end
```

第一个处理程序非常容易理解。但如果我们写了第二个处理程序后，把它存储起来，等过了一年以后，还能立即回忆起它的作用吗？那么一个 100行的处理程序又怎样呢？

China-pub.com
下载

除了直接注释外，还可以通过为处理程序和变量取一些有具体含义的名称，而达到程序易于理解的目的。例如，我们可以把一个处理程序命名为 ConvTemp。但如果把它的名称改为convertTemperature将会更好；该处理程序中可以有变量 f和c，但如果把它们的名称改为fahrenheitTemp和centigradeTemp，程序的可读性会更好。因为在 Lingo可以使用长的变量名，所以要尽量使用这个优势，使程序更加可读。由于在 Lingo剧本窗口里可以使用拷贝、剪切和粘贴的手段，就更没有理由使用短的、单个字符的名称了。

使用颜色、格式、注释和形象化的处理程序名和变量名，可以节省我们的时间，减少问题。从刚开始就习惯使用这些手法，我们的工作会更加顺利。

12.7 Lingo的故障排除

■ 请记住，当一个处理程序执行完后，其中的局部变量就不复存在了。到下一次该处理程序再次运行时，局部变量还需要重新赋值。如果想要变量能保持住它的值，应使用全局变量或行为属性变量(参见第14章“创建行为”)。

■ 即使你不打算使用所有的消息及事件处理程序，也应该熟悉它们。不能把事件处理程序的名称用作你的自定义的处理程序或变量的名称。也不能使用其他 Lingo的专有名词来作为自定义的处理程序或变量的名称。

■ 每个Lingo程序员都会遇到这样的问题：影片剧本不能按预计的那样运行。程序代码看起来都很完好 —— 剪辑室、演员表、舞台都很正常 —— 但剧本不起作用，好像不存在一样。这很可能是由于错误地把影片剧本设成了行为剧本。要想纠正这个错误，只要使用该演员的Script Cast Member Properties对话框把它设置成正确的剧本类型即可。

12.8 你知道吗

■ 你可以改变消息窗口内的文本的字体，方法是先选中文本，再使用文本监察窗。当你在教其他人学习Lingo，需要把消息窗口投影在屏幕上时，这一点尤其有用。

■ 由于消息窗口可以通过put命令进行数学运算，必要时可以使用它，而不必去打开另一个计算器软件。我觉得它很有用，比如可以用它来计算屏幕上两点的中点的坐标。

■ 可以用Lingo的scriptText属性得到某个剧本演员的文本，也可以用该属性定义某个剧本的文本。

■ 可以用Script Preferences对话框改变剧本窗口的缺省字体。也可以打开或关闭自动上色功能。如果关闭了自动上色功能，还可以用文本监察窗手工地为文本上色。

■ 处理程序的名称内可以包含各种符号及标点。例如，可以使用问号和惊叹号。可以为处理程序命名为on !。在变量名称中也可以使用各种符号。

■ 许多Lingo程序员习惯于在每个处理程序的结尾处的 end后面重复一遍该处理程序的名称。这在其他编程语言中是很常见的。它是写程序的一种方法， Director不需要这么做。