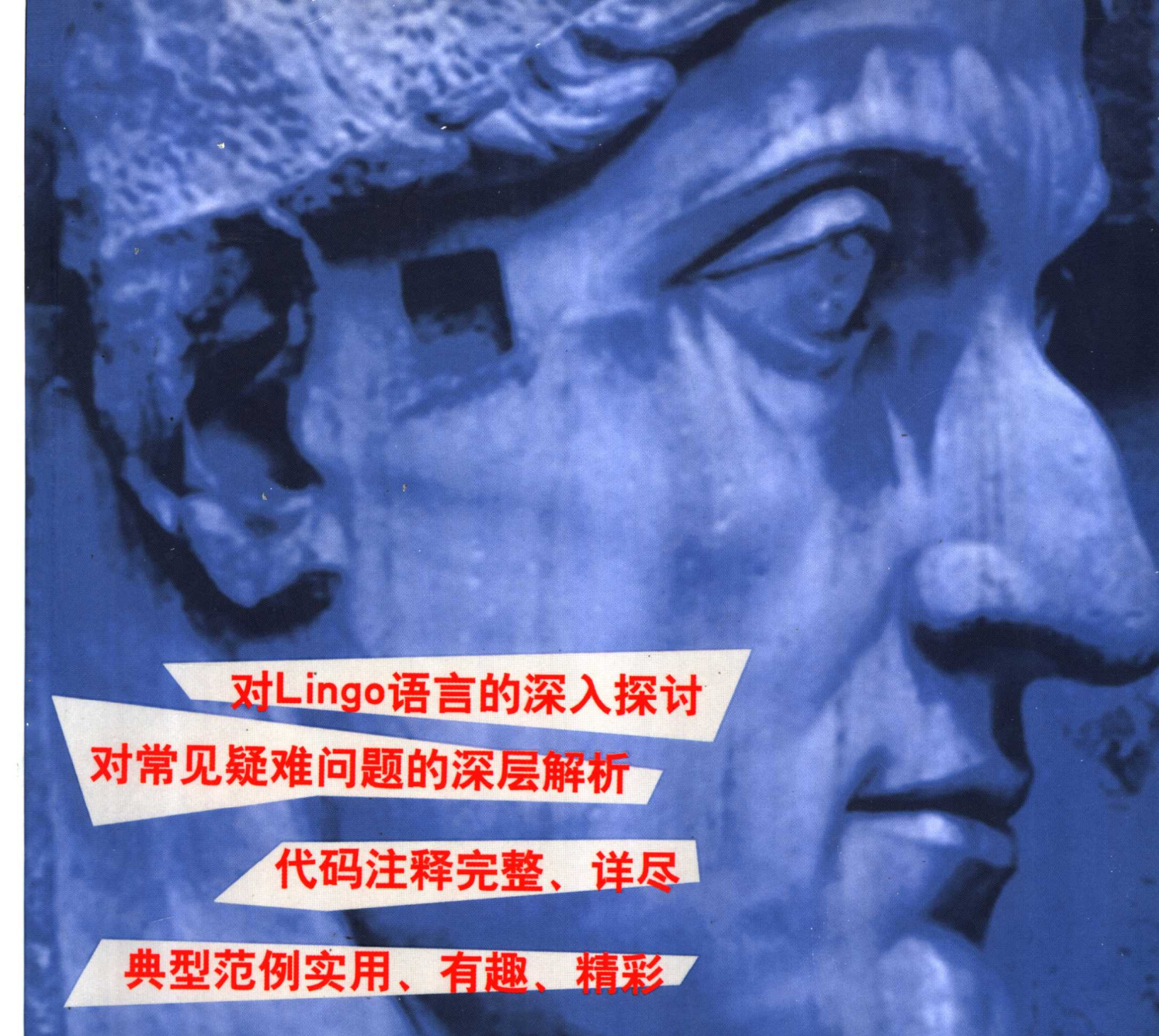




**对Lingo语言的深入探讨
对常见疑难问题的深层解析**

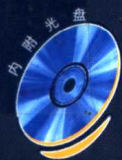
代码注释完整、详尽

典型范例实用、有趣、精彩

Director 8.5 Lingo

解析与实例

黄勇 宋博 冼枫 杨英华 编著



清华大学出版社

Director 8.5 Lingo解析与实例

黄 勇 宋 博
冼 枫 杨英华 编著

本书配有光盘，需要的读者请到 <http://210.34.51.1/tractate/index.asp>

网页上申请，或到“网络与光盘检索实验室”联系。

清华大学出版社
北 京

内 容 简 介

Director 是一套结合了多媒体制作和 Lingo 程序设计的一体化综合性软件, 它可以将多种不同类型软件制作出来的图片、文字、动画、声音、视频乃至虚拟场景等素材通过 Lingo 语言整合成为一体, 形成一个能够跨越平台播放的多媒体介质。其应用范围非常广, 在教学、产品展示、企业宣传、广告、游戏、网络传播等方面均具有巨大的实用价值。在当今的多媒体软件制作领域, Director 处于绝对的领导地位。

本书以必要的基础知识开篇, 对构成 Director 的几种主要分类元素, 进行了深入、全面、系统的实例剖析, 最后以综合实用的实例为总结。整个过程伴随着 Lingo 的深层解析, 从原理和思路展开探讨, 再加上各种实用经验技巧的介绍, 使读者能够在详尽的讲解中体会到 Lingo 的编程要领, 能够获得用 Lingo 对多媒体素材进行常用的控制以及构建完整多媒体程序的能力, 从而最终进入到一个新的多媒体创作层次, 并获得在多媒体行业从业的能力。

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: Director 8.5 Lingo 解析与实例

作 者: 黄 勇 宋 博 冼 枫 杨英华 编著

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](http://www.tup.tsinghua.edu.cn)

责任编辑: 庾 永

印 刷 者: 北京牛山世兴印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 44.5 字数: 1023 千字

版 次: 2003 年 4 月第 1 版 2003 年 4 月第 1 次印刷

书 号: ISBN 7-89494-040-2

印 数: 0001~4000

定 价: 63.00 元(附光盘)

序 言

Director 是美国 Macromedia 公司为高速发展的多媒体开发的一种核心软件。该软件同 Macromedia 的其他多媒体产品一样具有极其强大的竞争力，在当今多媒体开发市场上占有很大的比例。据统计，它在美国专业 CD-ROM 开发市场占据 85% 以上的份额。由于它不仅具备直观易用的用户界面，而且拥有很强的编程能力（内置强大的脚本语言 Lingo），近年来逐渐成为国内多媒体开发的主流工具。

Director 结合了多媒体设计和 Lingo 程序开发，将图片、文字、动画、声音、视频乃至虚拟场景等素材通过 Lingo 语言整合成为一体，形成一个能够跨越平台播放的多媒体介质。其应用范围非常广，在教学、产品展示、企业宣传、广告、游戏、网络传播等方面均具有巨大的实用价值。在当今的多媒体软件开发领域，Director 处于绝对的领导地位。

近几年来，随着多媒体电脑的普及，国内出现了越来越多的多媒体从业者和爱好者，其中不乏以 Director 为主要开发工具的群体，大家急切需要相关的 Director 辅导书籍。本书在 Director 8.5 版本基础上编写，通过对 Lingo 语言精练的讲解并结合典型范例，使读者深入了解 Lingo 语言，并能够获得切实的实践运用经验以及自我化解学习中的难点的能力。

本书旨在向已经掌握了 Director 基本操作、想进一步深入学习 Lingo 编程的人员提供学习 Lingo、吸取实例精华的机会；并培养读者学习计算机多媒体实用技术的兴趣与意识。在本书中，作者力求通过对实用技巧的透彻讲解和对 Director 制作中最常见疑难的深入解析，使读者能够在详尽的讲解中体会到 Lingo 的编程要领，获得用 Lingo 对多媒体素材进行常用的控制以及构建完整多媒体程序的能力，从而最终进入到一个新的多媒体创作层次，并获得在多媒体行业从业的能力。

本书从实用角度出发，具体讲解在 Lingo 编程中如何对素材进行完整地控制，以及教授构建完整程序的能力。主要体现在：实用技巧的透彻讲解和常见疑难的深入解析相结合。书中的精华来自目前国内最具专业性的 Director 网站——<http://www.aougu.net>，结合其专业论坛中的热门问题进行深入解析；面向有 Director 基础的人员，以 Lingo 为核心进行拓展，包括 Lingo 编程基础、多媒体素材控制专题、综合范例等。

本书既适合初涉多媒体开发领域的读者作为指导书籍，也适合已有一定开发经验的从业人员作为参考工具。

本书由 4 位多年从事多媒体工作的业内人士编写。4 位作者同时也是国内多媒体知名站点奥古多媒体 www.aougu.com 的 Director 版核心版主，他们在多媒体开发领域经验丰富，开发成果喜人。

奥古多媒体站长：岳永寿

2002 年 10 月 17 日

前言

近几年来,在国内已经出现了越来越多的 Director 相关从业者和爱好者,大家急切需要相关的辅导书籍。但是,纵观书架上的 Director 书籍,几乎全部集中在软件基本操作和 Lingo 语言基础知识的讲解上,这只能使读者达到一个初级的应用水平。在现有书籍中尤其缺乏的是对 Lingo 语言的深入探讨和范例分析,而这恰恰是 Director 的精髓所在。Lingo 作为 Director 的内嵌程序语言,已经将 Director 的优势发挥到极致,熟练地应用 Lingo 语言丰富和加强多媒体产品的设计能力,是作为一个 Director 多媒体程序员必备的从业能力。本书力求通过对 Lingo 语言精练的讲解并结合典型范例,使读者深入熟悉 Lingo 语言,并能够获得切实的实践运用经验,以及自我化解学习中的难点的能力。

一、编写目的和指导思想

本书旨在向已经掌握了 Director 基本操作、想进一步深入学习 Lingo 编程的人员提供学习 Lingo、吸取实例精华的机会;培养读者学习计算机多媒体实用技术的兴趣与意识。在本书中,我们力求通过对实用技巧的渗透讲解和对 Director 制作中最常见疑难问题的深入解析,使读者能够在详尽的讲解中体会到 Lingo 的编程要领,获得用 Lingo 对多媒体素材进行常用的控制和构建完整多媒体程序的能力,从而最终使读者进入到一个新的多媒体创作层次,并获得在多媒体行业从业的能力。

二、本书特点

本书从实用角度出发,具体讲解在 Lingo 编程中如何对素材进行完整地控制,以及教授构建完整程序的能力。主要体现在:实用技巧的渗透讲解和常见疑难的深入解析相结合。

(1) 采用目前最新版本的软件进行编写,并且尽可能采用国人的习惯。

(2) 本书中的精华来自目前国内最具专业性的 Director 网站: www.aougu.com, 结合其专业论坛中的热门问题进行深入解析。

(3) 面向有 Director 基础的人员,以 Lingo 为核心进行拓展,包括 Lingo 编程基础、多媒体素材控制专题、综合范例。兼以实用技巧以及疑难解答穿插书中各处。

(4) 本书中所有范例具有完整代码注释,详尽剖析。

(5) 在内容取舍、文字描述、版面布局等方面,力求适宜,让读者能够很顺畅地看懂、学会,行文中适当穿插“提示”、“注意”等,以丰富内容、活跃版面。

本书由黄勇、宋博、洗枫、杨英华编写。

由于时间仓促,书中如有不妥之处,恳请读者批评指正。

作者

2002 年 10 月

目 录

第 1 章 Lingo 语言基础.....	1
1.1 基础.....	2
1.1.1 Lingo 简介	2
1.1.2 编程基础.....	2
1.1.3 创建简单的 Lingo 脚本	24
1.2 列表.....	32
1.2.1 列表操作.....	32
1.2.2 属性列表.....	34
1.3 行为.....	35
1.3.1 认识行为.....	35
1.3.2 行为的结构.....	36
1.3.3 行为的本质.....	39
1.3.4 定制 Parameters 对话框	40
1.3.5 限制行为作用的精灵.....	43
1.3.6 向精灵发送信息.....	44
1.4 面向对象编程.....	47
1.4.1 面向对象编程概述.....	47
1.4.2 父脚本和子对象.....	48
1.4.3 行为和父脚本.....	58
1.4.4 继承性.....	59
1.5 代码编写.....	63
1.5.1 脚本的调试和优化.....	63
1.5.2 提高代码效率.....	67
1.5.3 良好的编程习惯.....	75
第 2 章 文本演员.....	77
2.1 基础知识.....	77
2.1.1 文本的引用.....	77
2.1.2 文本演员常用属性.....	87
2.1.3 段落、格式.....	93
2.1.4 文本格式化.....	96
2.1.5 滚动操作.....	110

2.1.6	文本常量.....	112
2.1.7	字体控制.....	116
2.2	范例.....	119
2.2.1	简易文本浏览器.....	119
2.2.2	文本数据库示例.....	137
2.2.3	简易文本取词.....	156
2.2.4	用户登录界面.....	161
第 3 章	图形控制.....	177
3.1	基础知识.....	177
3.1.1	图形的引用.....	178
3.1.2	图形演员常用属性.....	182
3.1.3	图形格式的选择和优化.....	189
3.1.4	图形的读取和存储.....	192
3.1.5	Image Lingo 简述.....	194
3.2	范例.....	204
3.2.1	滚动画布（一）.....	204
3.2.2	滚动画布（二）.....	208
3.2.3	鹰眼与放大镜.....	213
3.2.4	多张图片浏览（一）.....	218
3.2.5	多张图片浏览（二）.....	223
3.2.6	简易大场景图片浏览.....	233
第 4 章	Flash 演员.....	255
4.1	基础知识.....	255
4.1.1	Flash 的引用.....	256
4.1.2	Flash 演员的常用属性.....	258
4.1.3	常用 Flash 指令.....	260
4.2	范例.....	264
4.2.1	Flash 的全控制.....	264
4.2.2	Flash 与 Director 交互.....	281
第 5 章	声音控制.....	286
5.1	基础知识.....	286
5.1.1	声音的引用.....	286
5.1.2	声音演员的常用属性和控制方法.....	289
5.1.3	声音格式的选择和优化.....	300
5.1.4	声音的预加载和木偶化.....	302
5.1.5	线索点的应用.....	305

5.2 声音控制范例.....	311
5.2.1 声音的两种特殊引用.....	311
5.2.2 MP3 播放器.....	319
5.2.3 文本朗读者.....	339
第 6 章 视频控制.....	356
6.1 基础知识.....	356
6.1.1 视频的引用.....	361
6.1.2 视频演员常用属性.....	363
6.1.3 视频格式的选择和优化.....	366
6.1.4 线索点的应用.....	372
6.2 范例.....	374
6.2.1 多功能视频播放器.....	374
6.2.2 DirectMedia 视频播放器.....	393
6.2.3 QuickTime VR 控制.....	402
6.2.4 综合实例.....	405
第 7 章 窗口控制.....	412
7.1 基础知识.....	412
7.1.1 认识和使用 MIAW.....	413
7.1.2 窗口演员的常用属性和控制方法.....	416
7.1.3 窗口模式的选择.....	418
7.1.4 窗口的相关操作.....	419
7.1.5 窗口之间的交互.....	419
7.2 MIAW 范例.....	421
7.2.1 窗口的基本控制.....	421
7.2.2 不规则窗口的制作.....	426
7.2.3 可移动窗口的制作.....	432
7.2.4 系统窗口的调用.....	437
第 8 章 插件点滴.....	440
8.1 基础知识.....	440
8.1.1 插件 (Xtra) 浅谈.....	440
8.1.2 网页相关插件 WebXtra 简介.....	441
8.1.3 图形相关插件 PhotoCaster 简介.....	444
8.1.4 视频相关插件 DirectMedia 简介.....	448
8.1.5 其他常用插件.....	451
8.1.6 插件使用中的路径引用问题.....	453
8.2 插件范例.....	455

8.2.1	插件范例——打印插件	455
8.2.2	插件范例——数据库插件应用	462
8.2.3	插件范例——WebXtra 的应用	480
8.2.4	插件范例——BuddyAPI 实战	485
第 9 章	综合实例	491
9.1	寓教于乐——《中国华东、华北八省行政区划》 拼图游戏兼地理课课件	491
9.2	深入 OOP 编程——《墨子守城》小游戏	506
9.3	聪明宝宝妈妈教	520
9.3.1	功能一览	520
9.3.2	相关知识点	520
9.3.3	进程概述	521
9.3.4	界面流程	521
9.3.5	进程讲解	523
9.4	文本朗读编辑器	547
9.4.1	功能一览	547
9.4.2	进程概述	548
9.4.3	界面设计	550
9.4.4	代码解析	552
9.5	综合实例——热图编辑器	570
9.5.1	热图概念	570
9.5.2	热图编辑器	579
附录 A	奥古函数库 2.0 版	624
附录 B	奥古中华行为库列表	665
附录 C	插件相关信息罗列	675
附录 D	源文件目录导航	697

第 1 章 Lingo 语言基础

Director 绝对不是一个简单的动画和演示文稿制作工具，它可以制作一个完整的商用程序、教学软件、二维和三维的游戏、Web 应用平台。只要软件能够完成的事，Director 丝毫不会逊色，因为我们有 Lingo——Director 的编程语言存在。如今的 Lingo 已经非常成熟，资深的开发专家认为，未来的编程语言应该像积木一样容易使用，程序员的工作就是按照用户的需求搭建程序积木。可如今的 Visual C++、Delphi，即使是有组件、COM 的存在，也依然要面对巨大的脑力工作和开发成本。熟悉 Director 基本操作的人应该知道，Director 可以算作一个应用软件，而不是一个开发软件，因为可以用鼠标而不是键盘完成工作，文字、图形、声音、视频，只要用鼠标排列一下即可完成一个成形的软件。而对于其他开发语言可能需要调用一个 Windows 的 API 函数来完成的任务，在 Director 中，因为有了 Lingo 的存在，我们只需一个内置函数即可，而这个函数也可能因为频繁使用早已变得耳熟能详。

但 Lingo 并不是玩具语言，它同样具有面向对象（Object-Oriented）的特性。如继承，它是一种面向对象的编程语言（Object-Oriented Programming, OOP），OOP 使我们的程序只需要写一遍，就能够运用于多个对象。在 Director 中，OOP 就是把程序和对象结合起来，程序就是 Lingo 代码，对象就是文本、图形、声音和视频等演员。

单纯的使用 Director，可以让用户欣赏你的作品；加入 Lingo，还可以让用户使用你的作品。

在本章中，主要就以下几个大的方面对 Lingo 进行严密细致的剖析：

- Lingo 简介；
- 编程基础；
- 创建简单的 Lingo 脚本；
- 列表；
- 行为；
- 面向对象编程；
- 脚本的调试和优化。

1.1 基 础

1.1.1 Lingo 简介

Lingo 是一种编程语言,可以通过指令控制计算机的操作,可以通过 Director 的分镜表、演员表和舞台等,要求计算机按照我们的要求来进行操作。Lingo 和其他的程序语言一样,有自己的命令、函数和语句,这些代码都是用英文单词表示的。因为具有了 OOP 的特性,所以对 Lingo 程序代码的解读就像和计算机对话一样。同时,自己思考问题的过程就是编码的过程,可以夸张一点说,大脑里是怎么想的,就怎么写出来,因为 Lingo 比其他语言要好学得多。Lingo 是在 C 的基础上建立起来的,所以在很多地方都可以看到 C 的影子,如 if 语句、case 语句、new、me 等。和其他语言不同的是,一段 Lingo 代码是一种演员类型,这是由 Director 自身的特性决定的——Director 中的任何对象都是以演员的形式存在的。在其他的语言中,可能一段代码就是一个文件,而一段 Lingo 代码则是以一个演员的形式存在,它与图形、文本、声音和视频一样,都存在于 Director 的演员表之中,我们有时也把一个 Lingo 演员称为 Lingo 脚本。当然,有的 Lingo 演员需要放到分镜表 Score 中才会起作用。

Lingo 的功能非常强大。在 Lingo 出现之前,编写一个三维的游戏,创建一个虚拟的聊天室,几乎都成了骨灰级的程序员的专利。人们对他们和他们的程序都趋之若鹜。但 Lingo 的出现打破了这一垄断,在如今的 Director 演示中,不使用 Lingo 的演示已经很少了。虽然 Lingo 不是万能的,但相对于一些传统的方法来说,恰当地使用 Lingo 能够达到更好的效果。下面就是 Lingo 能够一显身手的地方:

- 商业演示:在幻灯、投影仪下显示商业数据,如各种表格、饼形图等;
- 课堂教学和课件:模拟的课堂知识讲解和操作,例如放大镜成像原理;
- 企业宣传光盘:宣传企业自身形象、产品的介绍等;
- 二维、三维游戏:RPG 游戏、DOOM 等都可以实现;
- 三维虚拟聊天室:构建一个虚拟的社会;
- MUS:多人对战游戏;
- 触摸屏演示系统:公共场合下的演示、查询系统,如邮局门口的邮政编码查询、商场的导购系统;
- 真实场景模拟:网上试衣间、模拟飞机驾驶室、模拟军事演习等。

1.1.2 编程基础

本节将首先探讨一些基础问题。一个人坐在计算机前写程序可以称之为编程,当多个

人坐在房间里一起编程时就成为开发了。编程是个人的劳动，开发是集体的劳动。要想成为一名合格的程序员，就必须掌握编程和开发这两个基础。

在 Director 中，编程的基础要求不同于其他开发工具。其他的开发工具需要我们对操作系统有一个比较透彻的理解，知道什么是窗口、什么是对话框、什么是菜单、什么是鼠标单击的事件，系统如何对用户操作进行响应等，对操作系统的运作要有一个大概的了解。而我们对 Director 编程的要求则更高一些，因为多媒体程序是构架在应用软件之上的，所以还必须对 Director 这个软件非常熟悉，但单纯的对软件的熟悉并非如想象中的那么简单。更多时候，我们还要考虑如何使程序的错误减至最少、内存占用率最低（虽然我们没有很好的办法对内存直接进行控制，但可以通过可见手段减少内存的占用率，如文本和图形的恰当选择，反锯齿的使用等）、如何对应用程序进行发布等。

门槛虽然摆在眼前，但并没有什么可怕的地方。因为，这些知识的入门和上手是很快的。下面各知识点是我们进行 Lingo 编程的基础，主要包括以下内容：

- 变量
- 表达式
- 语句
- 运算符
- 句柄与函数
- 脚本
- 事件
- 行为
- 属性

要想进行 Lingo 编程，首先应该对 Director 非常熟悉，至少知道什么是演员、什么是精灵、什么是舞台、什么是帧、什么是通道、什么是电影，以及它们各自的属性和彼此之间的区别与联系等。在掌握了这些基础后，学习好本章中介绍的编程基础知识，就可以进行实际项目的开发了。

在讲解编程的基础之前，有必要知道为什么要编程。即使是最简单的幻灯片演示，也需要有“上一个”、“下一个”按钮来控制演示的前进和后退。单纯的、流水线式的播放远不及一段视频动画来得精巧和吸引人。所以，增加演示的交互性能，更好地对现有素材进行安排控制，实现更多的要求，我们需要 Lingo 编程。

写程序是一项工程，不是所有人都能够编程，编程需要扎实的基础，需要对系统的深刻了解，需要对数学和物理的融会贯通。每个人的起点都不一样，因此，编程的基础对每个人而言也不是完全相同的。因为 Director 是一个集应用和开发为一体的软件，因而对我们的编程工作也就体现出了“艺术”和“技术”的双重要求。艺术，就是要求我们做出来的演示要有美感。Lingo 编程控制的是文本、图形、声音和视频。静态的演示，要求我们在舞台上设计排列出漂亮的图案，即使是最简单的一个按钮，我们也要让设计人员设计成特殊的图片。标准的 Windows 开发一切都遵循 Windows 的 GUI（图形化用户界面）要求，而我们则是尽可能地脱离 Windows 的标准化界面，尽可能地使用自己设计的图片来替代

Windows 界面元素，因为我们做的是多媒体演示，要让用户感觉这是一个艺术品，而不是一个程序；动态的交互要求我们体现用户的操作习惯，在准确的位置、恰当的时间显示正确的内容。技术的要求在此得到体现，Lingo 就是这样一个往来于艺术和技术之间的使者，将静态的、彼此独立的内容组织起来，使其成为一个有生命力的软件作品。

我们还必须具备以下几点要求，才能达到一个真正的程序员的要求：

- 规范化、标准化的编码习惯

从古到今，人们无时无刻不在以“不以规矩，不成方圆”来约束自己的行为。编码是一个高智力、高密度的脑力劳动，是一个循序渐进、逐步深化的过程。从变量的命名、注释的添加、模块的划分、算法的优化，到代码的简化、执行的效率，如果没有一个标准化、规范化的约束，必将成为一个混乱的线团，剪不断，理还乱，没有头绪、没有条理。一般来讲，变量命名普遍遵循的是匈牙利命名规则；注释的添加详细、必要却不重复；模块的划分有条理、紧凑，每个子模块只完成单一任务；算法力求绝妙、简洁、严密；代码的简化便于阅读分类，执行的效率力求速度最快和占用最小的系统资源，且有充分的冗错机制。试想 Linux 操作系统，由全世界的程序员来编写，如果每个人都是自成一体，全球的协作开发是不可想象的。

规范化、标准化的编码习惯具有以下几点好处：

- 便于自己的维护；
- 便于他人的理解；
- 便于比较和提高；
- 便于发布和交流。

- 编写文档习惯

这是程序员最不愿意做的事情，也是许多项目管理人员最为头痛的一个问题。为什么写一篇简单的文档就这么困难呢？文档有什么好写呢？还不及在程序中加的注释来得实在！很多人都会这么想。首先要知道什么是文档，文档不仅包括程序代码文档，还有需求分析文档、进度统计文档、演示（文件）描述文档、帮助说明文档、错误检测文档、用户操作说明文档等。单纯地在程序代码块的开头或者结尾添加大段的注释，应该是对该段程序的功能描述、编写时间、检测记录、备忘录、参数说明等。编写的文档应该为整个项目提供帮助，参考这些文档，下一个项目可以吸收经验教训，可以迅速从几十个、几百个演示中找到一个仅有三五行的函数。

- 需求理解能力

Lingo 程序员的需求理解能力来源于 3 方面：一是用户的需求；二是美术设计者的需求；三是代码的设计需求。用户的需求是笼统的、片面的、模糊的，甚至是错误的。鉴于中国国内目前的大多数情况，Lingo 项目没有专门的系统分析人员，没有专门的市场人员，大多是 Lingo 程序员直接面对用户，倾听他们对产品的要求。但用户并不了解技术，他只要求产品中有好听的音乐，可以让用户交互，可以连到互联网上。好听的音乐对程序员意味着什么：多少首音乐？循环播放？用户可以调节音量吗？打开新窗口时背景音乐会中断怎么处理？音乐采取什么格式？是直接导入 Director 中还是外部链接形式？所以 Lingo 程序员

必须具备良好的需求分析理解能力，能够将用户简单模糊的要求转换为具体的书面文字，一条一条地列举出来，每一条都能回答一个实际问题，而这些问题就是编程时需要解决的问题。相对而言，美术设计者的需求要简单多了，大多是如何按照美术设计者的要求来排列图形元素，如何以他的要求来制作动画等；代码的设计需求可以参考前面“规范化、标准化的编码习惯”一条。

- 团队精神和协作能力

要想告别个人英雄主义的单打独斗，要想告别作坊式的杯水车薪和鱼目混杂，必须具备团队精神和协作能力！软件是一个工程，是团体的成果。团队是一种精神，需要程序员具有主动性，需要一种宽阔的气魄，需要一种舍小家、为大家的精神；协作是一种能力，如何配合各种不同的人，如何寻找最好的协作方式，如何找到最好的沟通方法。协作体现了程序员做人的水平。

- 复用性、模块化的思维能力

人们总是说编程应该像搭积木一样简单，把现有的模块按照接口要求搭起来就可以用了，但前提是程序必须是一个模块。程序员们都抱着这个理想，却在自己的实际中不遵循这一理想。一段代码的复用性如何，能不能用在别的地方，能不能不受参数的影响等，都需要在编写之前进行深思熟虑，Java 可以写一次就能够到处运行，我们的代码模块就应该写一次后再也不用修改了。模块的划分原则一般是根据完成的功能进行的，例如，某个函数专门完成字母的大小写转换，某个函数专门得到指定目录下的文件列表，某个函数专门实现窗口的拖动等。有了模块化的程序，复用性的要求也就基本达到了。

- 测试习惯

无论何时何地，无论什么系统、什么人，都必须承认一点：世界上没有不出错的程序！我们习惯于将程序中的错误称之为臭虫（bug），捉虫也是程序员的必修课之一。无论什么时候都不要忘记寻找程序中的错误。作为 Director 的 Lingo 程序员，捉虫还包括寻找演示中的文字错误、图片上的错误等。测试一般包括两方面，一是正常使用测试，即程序能否在系统正常情况和用户正常操作下正常运行；二是异常测试，即在假设的系统有问题，用户错误操作下的测试，这一点最容易忽视，但也常常是返工最多的原因。

上面从软件开发的角提出了一些编程的要求，下面就具体针对 Director 的开发提出一些要求，这是进行 Lingo 编程的基础，掌握了这些知识，才能够让演示更加生动、更富有生命力。

1.1.2.1 变量

变量，就是一个变化的量，变量用来存储可变数据。在计算机中，变量占据着内存中的某一块区域。当程序退出时，程序会清空内存，于是，所有变量就消失了。在 Director 中，变量的使用并不需要事先声明，不像 C 语言那样，如果要引用一个整数型的变量 i，需要先写出 `int i` 语句，而在 Director 中，变量的出现是随时随地的。例如，可以把“张三”赋给 pName 变量，调用时直接写上 `pName = "张三"` 即可。不需要事先对变量的数据类型进

行声明。Lingo 会自动根据变量的赋值进行调整。如果上面的代码中 pName 被设定为字符串类型，再接着使用下面的代码则会出错：pAge = pName + 20，因为字符串和数字是不能进行数学运算的。对变量的赋值是通过等号“=”实现的。

提示：在 Director 8 之前的版本中，会经常看到 set 语句，如 set pName = "aougu"，这种写法也是正确的，省略掉前面的 set 后代码照样正确。现在我们多省略 set 单词。

在给变量赋值后，就可以对其进行调用和更改了。在 Message 窗口中输入下面的命令：

```
put pName  
-- "aougu"
```

这就表示该变量的值已经存在于内存中了。变量可以是任何字符、数字和下划线的组合。例如，下面的语句都表示变量：

```
myName  
thankyou  
at_7  
123abc
```

除变量外就是常量和 Lingo 的专用语句了，分析和举例如下：

- (1) 常量：如 PI，表示圆周率；
- (2) 各种函数名称，如 random()、draw()、stop()等；
- (3) 各种属性和名字，如 image、rect、text、member、stage 等；
- (4) 各种鼠标和按键事件名称，如 mouseUp、mouseEnter、mouseWithin、keyDown、keyUp 等；
- (5) 各种电影事件和帧事件，如 startMovie、prepareMovie、enterFrame、exitFrame 等；
- (6) Director 自己专门用于程序编写的语句：on、if、else、end if、repeat、end repeat、while、case、end case、otherwise、end、begin、go、to、the、put、me、new、not、or、return、do 等，这些语句通常被称为关键字。

在使用变量的时候，只要变量名称不和上面的相同即可。事实上，因为 Director 的 Script 窗口具有代码感应技术，如果是 Lingo 自带的单词，会以蓝色显示；如果是 Lingo 自带的函数或者语句，则会以红色显示；甚至可以自己定义关键字的颜色——在 File 菜单下执行 Preference 下的 Script 命令，然后在打开的对话框中进行设置即可。

在 C 语言中，变量具有不同的类型，这是为了更好地进行内存的管理，不同类型的变量占据的内存大小是不一样的。但在 Director 中，不用关心变量占据内存的大小，尽管使用即可，Director 会灵活分配。因为 Director 面对的用户群不是专业的程序员，有很多开发

者是非科班出生的，所以这极大地降低了开发的门槛！在 Director 中，变量主要分为数据型、字符串型、浮点型、布尔型 4 种。

(1) **数据型**：指整数，即 0、正数和负数。如果整数很大，Director 会以科学记数法来表示，如：

```
put 9999999999999999+3
-- 1.0000000000000000e16
```

(2) **浮点型**：指小数，无论正负。

对数字可以进行四则运算、三角函数运算。四则运算使用的符号是“+”、“-”、“*”和“/”，分别表示加、减、乘、除。

提示：乘号和除号的写法是为了方便计算机的输入，因为在键盘上找不到“×”和“÷”。具体使用方法请看下面的实例：

```
put 1+2
-- 3
put 9-5
-- 4
put 69*87
-- 6003
put 81/9
-- 9
put 81/10
-- 8
```

提示：在 Director 中，如果除数和被除数都是整数且除不尽的话，返回的结果也是整数，而不会是小数，且结果并不是四舍五入，小数部分将被截掉，只剩下前面的整数部分。为了得到小数，可以将除数和被除数其中任何一项变为小数（浮点数），例如：

```
put 81.0/10
-- 8.1000
```

或者使用 float() 函数进行转换：

```
put float(81)/10
-- 8.1000
```

(3) **字符串型**：以上两种类型外的变量都可以归类为字符串型。字符串型变量都是以引号形式来表现和引用的。文本中的内容也属于字符串，例如，要将字符串“我的名字是”

和 pName 这个文本型的 member 中的内容连接在一起，然后将其赋给 pShowName 这个字符串变量，相关代码为：

```
pShowName = "我的名字是" & pName.text
```

字符串变量总是带有引号的，字符串变量可以是空，即""（注意，中间没有空格，因为空格就表示一个空格符）。字符串的长度可以说是没有限制的，只要计算机允许。所以在很多情况下，会看到类似下面的代码：

```
myHtml = "<HTML>"
myHtml = myHtml & "<HEAD>"
myHtml = myHtml & member("cnmeta").text
myHtml = myHtml & "<TITLE>www</TITLE>"
myHtml = myHtml & "</HEAD>"
myHtml = myHtml & "<BODY>"
myHtml = myHtml & "<FONT SIZE=3>www.aougu.net/bbs 欢迎您拜访专业的 Director  
网站</FONT></BODY>"
myHtml = myHtml & "</HTML>"
```

上面的代码不断地将新字符串添加到原字符串后面，所以，到最后把一个网页的 HTML 源代码都赋给了一个变量 myHtml。然后，将变量 myHtml 中的字符串保存到 HTML 文件中，就生成了一个网页。所以，字符串的这种操作是很常见的，我们称之为字符串变量的叠加。

（4）布尔型：表示真或假，即判断结果，不是对就是错。在程序中可以用 TRUE 和 FALSE 或者 1 和 0 表示。布尔类型的使用场合多在 if 判断语句和函数的返回值中。

数据型和浮点型的引用是不带引号的，加上引号后就变成字符串了。看下面的语句，在 Message 窗口中输入下面的代码 `alert 48.4 + "12"`，按下回车键后将弹出如图 1-1 所示的对话框。

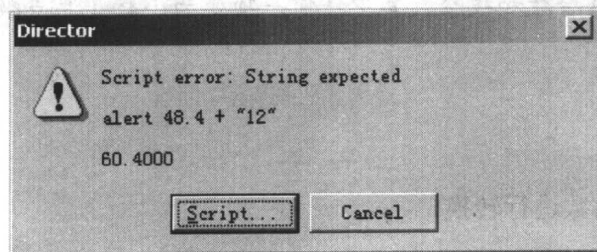


图 1-1 脚本错误提示

String expected 表示是数据类型不匹配错误，因为 alert 后面的参数必须是一个字符串，而 48.4 的出现却证明这不是一个字符串。但同时也会看到下面 60.4000 却是正确的运算结果。这就涉及到了变量类型的转换。