



微计算机丛书

Apple II

# LOGO程序设计

[美] 哈罗德·艾贝尔森 著

胡扬骅 译

电子工业出版社

1518/221  
411

Apple II  
LOGO 程序设计

[美] 哈罗德·艾贝尔森 著  
胡扬驿 译



电子工业出版社

8710515

## 內 容 簡 介

JS 150/15

LOGO是計算机輔助教学(CAI)中一种主要的語言。它易学易用、表达能力强,并有許多独特的功能,如表处理和图形处理等。特别适用于小学、中学和大学师生。目前在美国LOGO已經成为孩子們的第一計算机語言。LOGO很有可能成为取代BASIC的一种通用程序設計語言。

本書主要介紹LOGO系統的基本命令,过程使用,“龟图”和屏幕上龟图的基础知識,用数、字和表編写过程的方法,以及优秀的LOGO編程实例。

随着我国微計算机的普及,将会有愈来愈多的人使用LOGO語言。

LOGO for the Apple II

Harold Abelson

BYTE 1982

Apple II

LOGO 程序 设计

[美] 哈罗德·艾贝尔森 著

胡揚驛 译

責任編輯 王惠民

电子工业出版社出版(北京市万寿路)

新华书店北京发行所发行 各地新华书店经售

电子工业部情报所印刷厂印刷

开本: 850×1168 1/32 印张: 8.375 字数: 217千字

1987年3月第1版

1987年3月第1次印刷

印数: 9400册

定价: 2.15元

统一书号: 15290·160

## 序 言

Logo是一种计算机辅助教学语言，其宗旨在于让人们掌握对强有力的计算资源的个人控制，从而与来自科学、数学和智能模型结构艺术的深奥概念建立密切的联系。计算机语言是用来把计算机变成辅助学习、娱乐和研究的灵活工具。

Logo的设计者们的指导原则是，将Logo视为一种易于入门而又很少限制的教育工具。以至于即使是幼小的孩子，甚至于是头一次碰到Logo的人，都可以按自诱导的方式控制计算机。我们还相信，Logo应当成为一种有相当丰富表达能力的通用程序设计系统。事实上，我们觉得这两个目标是互为补充，并行不悖的。象BASIC一类的语言非常缺乏基本语言的表达能力，所以初学者很难用它编写能完成许多有趣任务的简练程序。积MIT(Massachusetts Institute of Technology——美国麻省理工学院)和各地逾十年之久的经验，具有“数学素养”的人们都喜欢用Logo来生成新颖而精巧的程序。Logo广泛而成功地适用于初小、高小、中学和大学的师生。Logo的主要特性如下：

Logo是一种过程语言。通过把命令组合成组——过程——生成程序。然后又把这些过程作为另一些过程的过程步，并可照此继续下去组合成任意层次的复杂结构。过程的每一个独立的过程步可以是Logo基本命令或用户定义的过程。过程之间经过输入和输出进行通讯。

• Logo是一种交互作用编程语言。任何Logo命令，无论是在语言中还是定义成为过程，都可以在键盘上简单地打命令键执行。Logo的集成编辑器很容易对过程进行定义、执行和修改，因为没有必要对编辑器和监控器等分别加以处理。

• Logo的数据结构（即可以用变量命名，直接作为过程的输入，并作为量值返回的数据体）不仅包括数和字符串，而且包括一种称之为表（list）的混合结构。许多计算机语言都迫使程序员按照在特殊的数和串上所进行的操作顺序来管理数据结构。与此相反，Logo的表可以作为单独操作的功能单元，因而使其成为应用于符号处理的一种方便而强有力的语言。而且，Logo过程本身可以表示和处理为表的事实表明，通过对命令加以翻译，为诸如生理上有缺陷或非常年幼的人提供Logo特殊接口的方式，可使用户取得相当直接的控制。

Logo的另一重要特性是名为“龟图”（turtle geometry）的编程区。龟是一种计算机控制的“模拟动物”，出现在显示屏幕上，响应使其移动（前或后）和旋转（左或右）的Logo命令。当龟移动时，在屏幕上留下其运动轨迹，由此可用于在屏幕上绘图。例如，下面的Logo过程说明了如何通过重复四次命令：“向前100个单位，然后向右转90°”，使其绘出一个矩形。

```
TO SQUARE  
  REPEAT 4 [FORWARD 100 RIGHT 90]  
END
```

龟图无论是作为各种年龄的人的程序设计入门教程，还是作为以计算机为基础的数学课程都是极为成功的。本书中我们使用龟图引入Logo程序设计的基本概念，尽管也包含语言的其它方面的内容。

### 微型计算机的Logo

Logo自1968年创立以来，已经有了持续不断的发展。因为Logo是一种高级的语言，因而七十年代绝大多数的Logo研究工作都是在大型计算机系统上实施的。只是到了今天，能够支持Logo的计算机变得如此便宜，使其足可以扩大到学校和家庭的应用中。1979年，MIT的Logo小组开始了使Logo适用于这些机器的努力，并产生了两种实施方案：一种是用TI99/4个人计算机，另一种是

Apple II个人计算机。本书中的实例取材于Apple实施方案，但Logo的这两种版本是很相近的。我们希望Logo的这些实施仅仅是众多实施的开端，一种简便而有力的编程语言的普遍适用性，将有助于加深每一个人理解和控制计算机的能力。这一点亦正是其力量之所在。

### 阅读本书的指导

本书是对Logo系统和用Logo进行程序设计的导论。你可以将学习Logo分为三步。第一步包括第一、二章，介绍定义过程和使用龟图在屏幕上绘图的基础。第三章是在此基础上对编程方法的建议。第四章介绍了跟踪过程并将其存入磁盘文件的方法。第二步包括用“数据”——第五章中介绍的数、字和表——编写过程和实现在第六、七章中所介绍的那些设计。第五章的最后一节还讨论了Logo句法的某些长处，这些在头四章中几乎被忽略了。第八、九章中包含用Logo编程的若干优选的题目，其中包括采用递归处理字和表及用表来代表复杂的数据结构。第十章是介绍的Logo系统基本命令的汇总。

本书中所采用的实例，来源于从事教育研究的MIT分部和MIT人工智能实验室的Logo小组成员过去十多年的研究成果。第6.3节中重印了由Seymour Papert和Cynthia Solomon合写的，发表于1970年的AI Lab Memo上的文章。

第三章中的设计取材于Dan Watt为马萨诸塞州布鲁克林市的小学系统所实施的教学实验部分而准备的材料。我要感谢Greg Gargarian为编排这一章所给予的帮助。第七章中的Dynaturtle的设计是基于Andy disessa和Dan Watt的工作。我还要对Dan Watt, Leigh Klotz, Nola Sheffer及Richard Carter为本书的前期草稿所作的注释表示感谢。

# 目 录

第一章 Logo入门.....	(1)
1.1 使用Logo前的准备.....	(1)
1.2 Logo命令.....	(2)
1.2.1 基本图形命令.....	(5)
1.2.2 更正键入错误.....	(7)
1.2.3 出错消息.....	(9)
1.2.4 使用命令的实践.....	(10)
1.3 过程导论.....	(12)
1.3.1 简单过程.....	(13)
1.3.2 定义过程.....	(15)
1.3.3 过程中的错误.....	(20)
1.4 其它图形命令.....	(22)
1.4.1 用颜色绘图.....	(23)
1.4.2 控制绘图反转.....	(25)
1.5 屏幕使用模式.....	(25)
1.5.1 非绘图模式.....	(25)
1.5.2 编辑模式.....	(26)
1.5.3 绘图模式.....	(26)
第二章 用过程编程.....	(28)
2.1 过程的输入.....	(28)
2.1.1 多个输入.....	(31)
2.1.2 作为专用名的输入.....	(32)
2.1.3 圆弧过程.....	(35)
2.2 重复和递归.....	(41)
2.2.1 递归的探讨.....	(43)

2.2.2	条件命令和 STOP	(45)
2.2.3	递归的进一步探讨	(47)
2.2.4	画树	(51)
第三章	用龟图进行设计	(56)
第四章	工作空间、文件管理及调试	(77)
4.1	工作空间管理	(77)
4.1.1	显示 (PRINTOUT)	(77)
4.1.2	抹除 (ERASE)	(78)
4.1.3	EDIT的其它用法	(78)
4.2	Logo 文件系统	(79)
4.2.1	磁盘文件	(80)
4.2.2	存图	(81)
4.2.3	文件系统新用法	(82)
4.3	获得硬拷贝	(83)
4.4	辅助调试	(83)
4.4.1	跟踪 (TRACE)	(83)
4.4.2	暂停 (PAUSE) 和继续 (CONTINUE)	(85)
第五章	数、字和表	(87)
5.1	数和算术运算	(87)
5.1.1	指数运算符	(89)
5.1.2	整数运算	(89)
5.2	输出	(90)
5.3	字	(94)
5.4	表	(97)
5.5	命名	(101)
5.5.1	局部名和全局名	(103)
5.5.2	自由变量	(105)
5.6	条件表达式和谓词	(106)
5.7	关于Logo文法的细节	(110)
5.7.1	Logo怎样将行分割为字	(111)



5.7.2	圆括弧的用法	(112)
第六章	用数、字和表进行设计	(117)
6.1	算术测验程序	(117)
6.2	随机句(Random-sentence)发生器	(119)
6.3	Nim: 一种游戏程序	(123)
6.3.1	子目标设计	(124)
6.3.2	简单记分员	(126)
6.3.3	机械运动员	(129)
6.3.4	精练和修改	(132)
6.3.5	NIMPLAY过程的清单	(134)
第七章	编写交互作用程序	(135)
7.1	控制屏幕输出	(135)
7.2	键盘输入	(136)
7.2.1	范例: 对小孩子的立即响应	(137)
7.2.2	键盘控制运行中的进程	(138)
7.3	范例: 动龟程度	(140)
7.3.1	什么是动龟?	(140)
7.3.2	动龟的运动	(142)
7.3.3	改变动龟的运动	(143)
7.4	从操纵杆(paddle)输入	(145)
第八章	输入、输出和递归	(147)
8.1	颠倒字和表	(148)
8.2	处理表的递归过程	(155)
8.2.1	LENGTH过程	(155)
8.2.2	PICK过程	(158)
8.2.3	MEMBER? 谓词	(160)
8.3	进制转换	(162)
第九章	表的新用	(165)
9.1	层次结构	(166)
9.1.1	表算符	(167)

9.1.2 范例: 关联表 .....	(171)
9.2 象数据一样的程序 .....	(174)
9.2.1 RUN 命令 .....	(175)
9.2.2 DEFINE 命令 .....	(178)
9.2.3 TEXT 命令 .....	(182)
9.2.4 增加新的程序设计结构 .....	(183)
9.3 用表作更多的设计 .....	(185)
9.3.1 范例: DOCTOR 程序 .....	(185)
9.3.2 范例: ANIMAL 程序 .....	(189)
第十章 Logo 基本命令汇总 .....	(201)
10.1 图形命令 .....	(201)
10.2 数值运算 .....	(205)
10.3 字和表的运算 .....	(208)
10.4 过程定义和编辑 .....	(211)
10.5 条件表达式 .....	(212)
10.6 随条件表达式使用的谓词 .....	(214)
10.7 过程执行控制 .....	(215)
10.8 输入和输出 .....	(216)
10.9 命名 .....	(218)
10.10 文件和工作空间管理 .....	(219)
10.11 辅助调试 .....	(221)
10.12 编辑命令 .....	(221)
10.13 其它控制字符 .....	(223)
10.14 其它命令 .....	(224)
10.15 错误消息 .....	(223)
附录 I Logo 的 Apple Logo (TM) 实施方案 .....	(231)
附录 II Logo 的 TI Logo (TM) 实施方案 .....	(238)

# 第一章 Logo 入门

本章介绍使用Logo的基本方法，叙述了如何执行简单的命令和如何定义及编辑过程，并举了用龟图在屏幕上绘图的范例。尽管此时尚未介绍更多的命令，也未对写程序的规则作充分的解释，但本章及下一章的内容，足可以使你把Logo用于象第三章中所介绍的形形色色的有趣设计。应该在计算机键盘上把本章所介绍的内容都试一试，对每一种不同的特性都进行一番实验。

## 1.1 使用Logo 前的准备

如果你以前从未使用过计算机，就必须熟悉和比较计算机键盘和打字机键盘的不同点。必须留心，不要把数字0打成了字母O，或者把数字1打成字母l。这些键在人看来似乎差别不太大，但对计算机来说，却会被翻译成截然不同的信息。计算机键盘一般还有几个在打字机上通常所没有的键，在Apple II键盘上，有一个标有CTRL (Control) 的键，类似于SHIFT键用来打所谓控制字符。即若要打字符“Control G”，应在按下并保持CTRL键之后打G键（而不是试图同时打CTRL和G）。本书从头至尾，我们均指定前缀“CTRL”代表控制字符，就象在“CTRL-G”中一样。Apple的其它一些特殊键是ESC, REPT, RESET, RETURN和移格键。Logo还使用左和右方括弧字符〔和〕，这些字符未标在键上，而是用打SHIFT-N和SHIFT-M实现的。图1.1是Apple II的键盘及Logo使用的特殊键的示意图。关于键盘的详细使用，

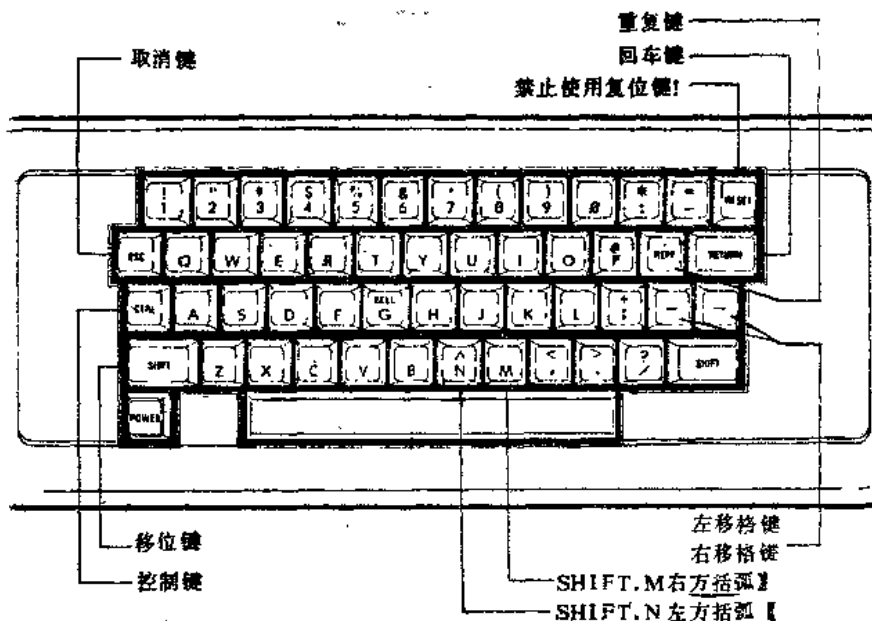


图1.1 标明特殊键的Apple I键盘

请参考Apple Logo 技术手册\*,该手册也介绍了如何在Apple 上装人和启动Logo系统的指令。

## 1.2 Logo 命令

图1.2是进入Logo时Apple屏幕的照片,系统显示出欢迎消息,随后在下一行由一个问号打头。该问号是提示符,表明Logo

\* 当使用Apple Logo时,应记住的最重要的一点是,千万不要打RESET键,这会破坏Logo并返回到Apple 监控器。若有可能,应在使用Logo时使RESET键不起作用。

正等待你发出的命令。

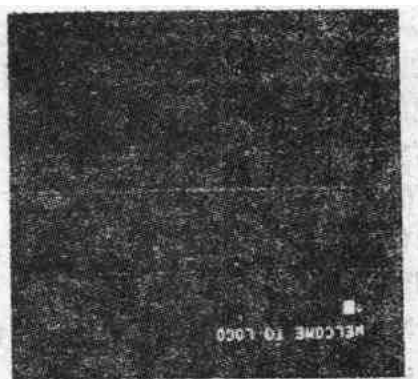


图1.2 Logo初次启动时的Apple屏幕

要向Logo发出命令，必须打入命令和RETURN键。例如，要让Logo显示36与37的积，应打入下述命令行：

```
PRINT 37 * 67
```

亦即打入键P、R、I、N、T、空格键、3、7、空格键、\*、空格键、6、7、RETURN键。计算机随即显示出2479，并另起一行以问号提示符表明已准备好接受新的命令。牢牢记住：当你打入命令行时，一直要等到打RETURN键时，该命令才被执行。为了告诉Logo显示消息：“Logo is a computer language”，应打入下述命令行：

```
PRINT [LOGO IS A COMPUTER LANGUAGE] 然后打RETURN键。这个例子说明在Logo中如何使用方括弧把字组合成为表 (list)。左和右方括弧字符在Apple键盘上分别对应于SHIFT-N
```

\* 既使没有空格分隔时，Logo也知道如何将一行划分成若干个组成元素。例如，它完全能够把由5个字符组成的串37 \* 67翻译成由三个元素组成：即数37，算符 \* 和数67。尽管如此，还是应养成一种良好的习惯，始终记住用空格把命令行中的元素分隔开，关于何处必须使用空格的规则在第3.7节中讨论。

和SHIFT-M。可以用这种类型的表在屏幕上显示消息，除此而外，表还有别的用途，我们将在第五、九章中详细介绍。

命令行中的空格很重要，因为它向Logo表明，一行命令是如何分成若干组成部分的。如果在打上述头一个命令时，省略了T和3之间的空格变为：

```
PRINT37 * 67
```

那么Logo将以为你让它执行名为PRINT37的命令，却又不知道怎样去执行它，因而用下述错误消息回答：

```
THERE IS NO PROCEDURE NAMED PRINT37
```

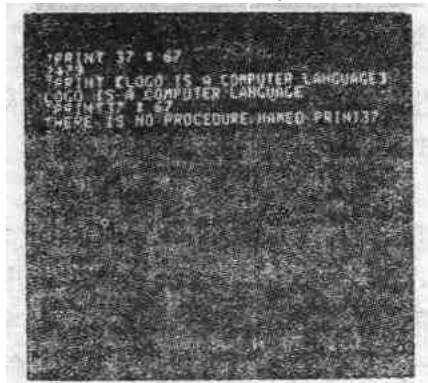


图1.3 向Logo打入的三行命令及系统的回答

图1.3是当你打入上述三行命令，并得到计算机系统对每一行的回答时的屏幕照片。每行命令开头的问号是由Logo产生的提示符，其它部分则是由使用者打入的命令。在本书中，当我们要特别强调区别使用者打入的字符和Logo产生的字符时，就把后者用斜体形式显示。例如：图1.3中的第一个命令将显示为：

```
? PRINT 37 * 76
```

```
2479
```

在以后各章中，将会看到如何编写处理数和文本的Logo程序。但刚开始学习Logo时，主要是研究通过向“造图者”即龟图

发命令，如何让计算机在显示屏幕上绘图。为了建立绘图用的屏幕，打DRAW和RETURN键，这时屏幕显示如图1.4所示，整幅屏幕呈黑色，中央有一个小的白色三角形，接近屏幕下方有一个白色问号。该问号跟先前提到过的一样，作为提示符表明Logo已作好接受命令的准备。作图的时候，Logo保留屏幕下方的四行作为显示打人的命令和计算机的回答，屏幕的其它区域供绘图用。当Logo用这种方式在屏幕上绘图时，系统这时叫做绘图模式。如图1.2和1.3所示，屏幕上原先那种没有为图形保留空间的方式叫做非绘图模式\*。无论何时，只要Logo是在绘图模式，都可以用命令NODRAW使其回到非绘图模式。

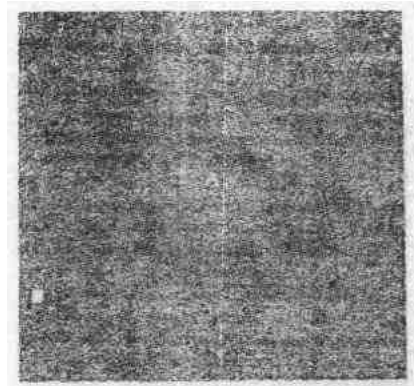


图1.4 当Logo进入绘图模式时Apple屏幕的样子

### 1.2.1 基本图形命令

龟图是Logo进入绘图模式时出现在屏幕中央的三角形指针。通过命令龟图移动并留下运动轨迹进行绘图。移动龟图有四种命

---

\* 在非绘图模式时，Apple屏幕有24行供显示键盘打入的内容。关于Logo的不同工作模式的细节在第1.5节中解释。

令, FORWARD和BACK使其沿指针的方向移动, 每当你打入 FORWARD或BACK命令时, 还必须指定其移动的距离数。RIGHT和LEFT使其转动。每次打入RIGHT或LEFT时, 还必须指定其转动的角度数。如果打入下述Logo命令序列

```
RIGHT 45  
FORWARD 100  
LEFT 135  
FORWARD 200
```

将生成如图1.5所示的楔形图。要记住, 每一行命令均应以RETURN结尾, 在命令字和数字之间应留有空格。如果错打了一个字符, 可打ESC键删除那个错打的字符, 详见1.2.2节有关更改错打键的说明。

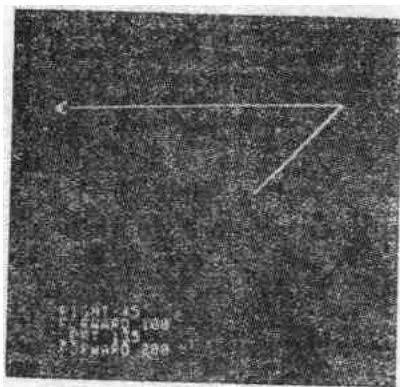


图1.5 显示一个简单的龟图命令序列的屏幕照片

紧跟在命令后面的数叫做输入。每个 FORWARD, BACK, LEFT和RIGHT命令都需要一个输入。Logo命令可能需要也可能不需要输入, 这取决于命令本身。DRAW就是一个不需要输入的例子, 以后, 我们还将看到需要不只一个输入的命令例子。

如果你要移动龟图, 但又不想画出移动轨迹, 应打PENUP命令, 接着打FORWARD或BACK命令。如果要恢复绘图功能,



应打PENDOWN命令。PENUP和PENDOWN均不需要输入。HIDETURTLE命令使龟图指针消失，虽然龟图仍旧还“在”，且如果画笔放下的话（PENDOWN）也能画出直线来。SHOWTURTLE命令使指针重现。图1.6是用上述命令绘制简单图形的说明。

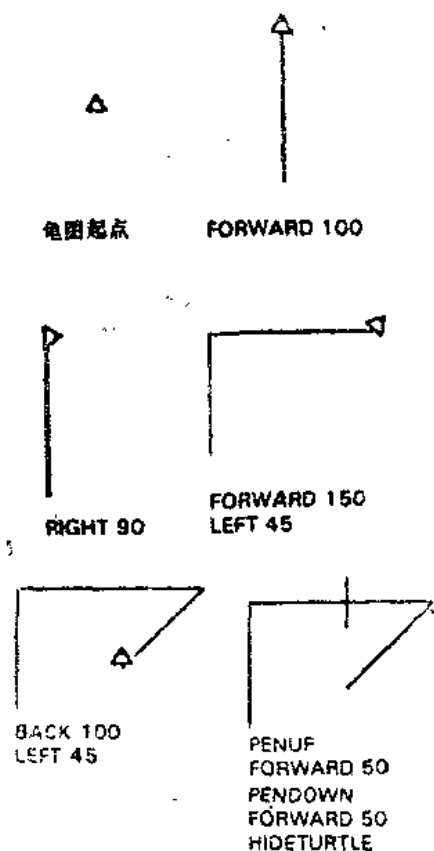


图 1.6 用龟图绘图

### 1.2.2 更正键入错误

当打入Logo命令时，有时会打错一些键。经常出现的错误是