

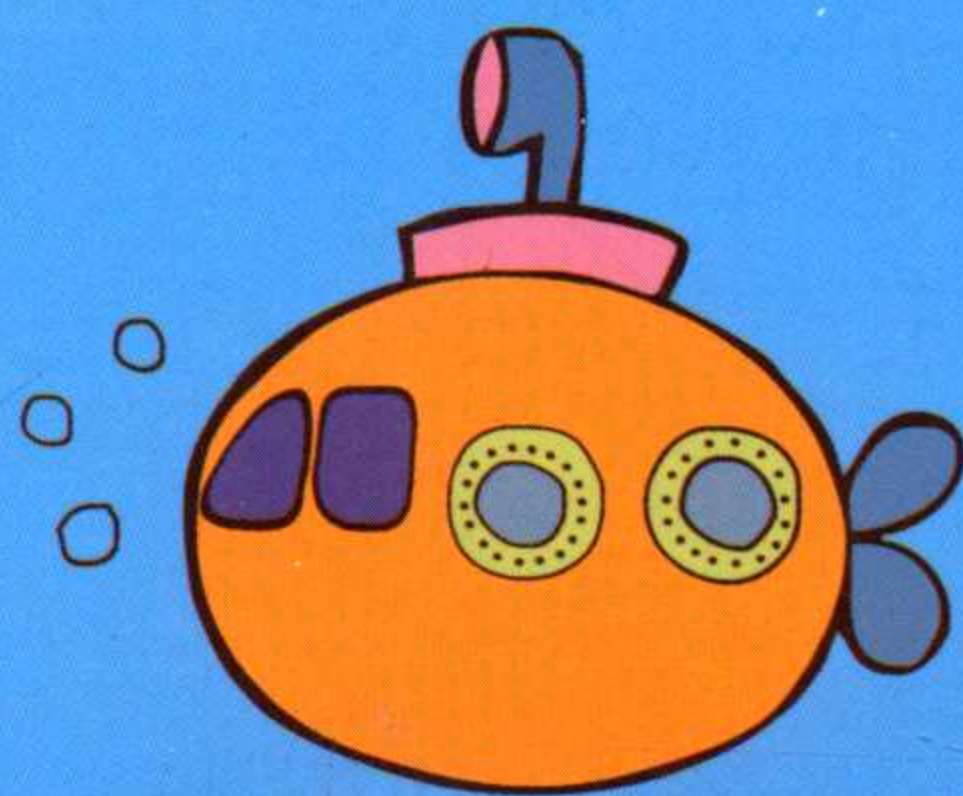
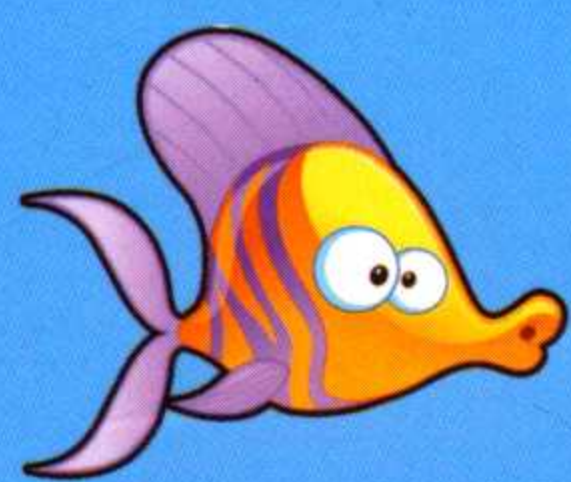
一本融合数学、游戏和算法的
青少年编程教材

Scratch

编程从入门到精通

—— 数学 · 游戏 · 算法 ——

© 谢声涛 编著



清华大学出版社



Scratch

编程从入门到精通

◎ 谢声涛 编著

清华大学出版社
北京

内 容 简 介

本书是全面讲解图形化编程语言 Scratch 的编程指南,分为基础编程篇、图形编程篇和进阶编程篇 3 个部分,系统讲解 Scratch 基础编程知识和运动、外观、侦测等各功能模块的技术应用,以及常用的数据结构和算法知识,并提供丰富有趣的教学案例和练习题,为青少年提供一本从入门到精通的 Scratch 编程参考教材。

本书假设读者从未接触过编程,从零基础开始帮助读者逐步建立起 Scratch 编程的知识体系。本书适合对编程有兴趣的青少年阅读,也适合希望辅导孩子进行编程训练的家长和少儿编程培训机构的教师使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Scratch 编程从入门到精通/谢声涛编著. —北京:清华大学出版社,2018
ISBN 978-7-302-50837-3

I. ①S… II. ①谢… III. ①程序设计—青少年读物 IV. ①TP311.1-49

中国版本图书馆 CIP 数据核字(2018)第 173127 号

责任编辑:王剑乔

封面设计:刘 键

责任校对:袁 芳

责任印制:董 瑾

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京博海升彩色印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:15.25

字 数:341 千字

版 次:2018 年 8 月第 1 版

印 次:2018 年 8 月第 1 次印刷

定 价:89.00 元

产品编号:077405-01

前言

Scratch 是由 MIT 媒体实验室为青少年开发的图形化编程工具,已被翻译成 70 种以上的语言,在超过 150 个国家和地区被广泛使用。就像玩乐高积木一样,使用 Scratch 编程简单而有趣。只要用鼠标从 140 多个不同功能的指令积木中选择和拖曳,把不同的指令积木按照某种逻辑关系拼搭在一起,就能得到一个可运行的程序,从而创建出各种交互式故事、动画、游戏、音乐和美术作品等。

Scratch 能够与数学、物理、语文等众多学科融合在一起,对青少年的学习有着非常大的帮助。例如,在 Scratch 中,不仅能接触到基本的算术运算、关系运算和逻辑运算,还能接触到平面直角坐标系、绝对值、平方根、三角函数等初等数学知识。毋庸置疑,游戏总是能够吸引青少年的注意力。用 Scratch 编写不同类型、不同复杂度的游戏时,需要适当地运用不同的数学知识来设计游戏的算法。例如,通过圆的参数方程来控制角色作圆周运动,通过抛物线方程来模拟炮弹的运动轨迹,等等。在游戏的驱动下,这些数学知识将不再枯燥乏味,将会驱使青少年主动探究这些在游戏程序中发挥关键作用的“秘笈”。通过“玩中学”,能够激发青少年主动学习和运用各个学科的知识。

学习 Scratch 编程最大的益处就是能够激发青少年的创造力。Scratch 简单易用且功能强大,能快速地将青少年的创意落地,变成一个个交互感极强的作品。在动手创作的过程中,根据项目的不同,需要融合 Science(科学)、Technology(技术)、Engineering(工程)、Arts(艺术)、Mathematics(数学)等多个领域的知识。例如,对于一个稍复杂的游戏项目,就需要策划、美工、编程等不同角色的人员参与组成一个开发小组,以团队协作方式共同创作项目。可以说,Scratch 编程是近年来流行的 STEAM 教育理念的一个极佳实践方式。

本书分为基础编程篇、图形编程篇和进阶编程篇 3 个部分。

第 1 部分讲授 Scratch 基础编程知识,介绍如何使用 Scratch 编辑器开发项目,如何使用变量、运算符、列表和过程等进行编程,如何使用流程图描述算法和进行结构化程序设计,等等。该部分提供丰富多彩的趣味数学案例,有韩信点兵、鸡兔同笼、数字黑洞、约瑟夫环、逻辑推理……

第 2 部分讲授 Scratch 图形编程知识,介绍如何控制角色在舞台上运动和进行碰撞侦测,如何更改角色的外观和创建各种特效,如何播放声音和模拟乐器演奏,如何使用画笔和图章在舞台上绘制图形和图案,等等。本部分以一个汇集诸多图形编程技术的“海底探险”趣味游戏贯穿各章,还引导读者创作电子相册、巡线甲虫、手势抓蝴蝶、种蘑菇、模拟乐器等趣味小游戏。

第3部分讲授 Scratch 进阶编程知识,介绍如何使用克隆功能和消息机制简化程序开发,如何编写和组织规模较大或功能复杂的应用程序项目,以及面向对象程序设计和事件驱动编程思想等。此外,还通过“英汉词典”和“走迷宫”项目介绍常用算法和数据结构的应用。

本书中的程序基于 Scratch 2.0(版本号为 v458.0.1)编写,所有范例程序均已调试通过。读者可以关注微信公众号“小海豚科学馆”获取本书的范例程序文件和素材,还可以加入本书的官方 QQ 群 126170677 获取学习资源,并可与本书编著者和网友进行在线交流。

本书假设读者从未接触过编程,从零基础开始介绍 Scratch 编程知识,帮助读者逐步建立起 Scratch 编程的知识体系,适合对编程有兴趣的青少年阅读,也适合希望辅导孩子进行编程训练的家长和少儿编程培训机构的教师使用。

由于水平所限,本书难免有疏漏或不妥之处,还请读者朋友不吝赐教。

谢声涛

2018年3月



本书配套资源下载

目 录

第 1 部分 基础编程篇

第 1 章 编程起步 3

1.1 Scratch 编程概述 3

1.1.1 为什么用 Scratch 编程
..... 3

1.1.2 从 Scratch 到 Python ... 5

1.1.3 Scratch 的主要版本 ... 7

1.1.4 Scratch 项目概述 9

1.2 准备 Scratch 开发环境 10

1.2.1 使用 Scratch 在线
编辑器 10

1.2.2 安装 Scratch 离线
编辑器 12

1.2.3 设置 Scratch 编辑器
..... 14

1.3 Scratch 编辑器界面 16

1.3.1 舞台区 17

1.3.2 背景和角色列表区 ... 17

1.3.3 脚本编辑区 18

1.3.4 造型和背景编辑区 ... 18

1.3.5 声音编辑区 19

1.3.6 菜单栏和工具栏 ... 20

1.4 Scratch 指令系统概述 20

1.4.1 积木的形状特点 ... 20

1.4.2 积木的操作方法 ... 22

1.4.3 积木的功能类别 ... 24

1.4.4 快速查看积木说明 ... 25

1.5 临摹案例 26

1.5.1 数学编程：韩信点兵

..... 26

1.5.2 游戏编程：星际飞行
..... 30

第 2 章 变量和运算 36

2.1 说和思考指令 36

2.1.1 跟我做：小猫背唐诗
..... 37

2.1.2 让角色说话和思考 ... 38

2.1.3 动手练：小猫的哲学
思考 39

2.2 问答指令 40

2.2.1 跟我做：简单的人机
对话 40

2.2.2 询问与回答 40

2.2.3 动手练：说出字母数
..... 41

2.3 变量 42

2.3.1 跟我做：小猫变大
变小 42

2.3.2 变量显示器 43

2.3.3 变量的数据类型和操作
..... 43

2.3.4 变量的命名和作用域
..... 45

2.3.5 动手练：求两数之和
..... 45

2.4 数学运算 46

2.4.1 跟我做：鸡兔同笼 ... 46

2.4.2 算术运算和运算优先级
..... 48

2.4.3	数学函数	48	第 4 章 列表	74
2.4.4	动手练：计算大桥长度	49	4.1 列表显示器	74
2.5 随机数	50		4.1.1 跟我做：求平均气温	74
2.5.1	跟我做：小鱼逍遥游	50	4.1.2 列表显示器的使用	76
2.5.2	生成随机数	51	4.1.3 动手练：评委打分	77
2.5.3	动手练：随机投点 画圆	52	4.2 列表的操作	78
2.6 字符串函数	53		4.2.1 跟我做：银行叫号系统	78
2.6.1	跟我做：成语接龙	53	4.2.2 列表操作积木	79
2.6.2	字符串处理	54	4.2.3 动手练：约瑟夫环	83
2.6.3	动手练：回文诗	55	第 5 章 过程	85
第 3 章 程序控制	57		5.1 自定义过程	85
3.1 顺序结构	57		5.1.1 跟我做：计算圆的面积	86
3.1.1	跟我做：海伦公式	57	5.1.2 使用自定义过程	87
3.1.2	流程图	59	5.1.3 动手练：判断质数	88
3.1.3	顺序结构的程序设计	59	5.2 递归的使用	89
3.1.4	动手练：计算圆锥体积	60	5.2.1 跟我做：辗转相除法	89
3.2 选择结构	61		5.2.2 递归调用的分析	91
3.2.1	跟我做：求绝对值	61	5.2.3 动手做：斐波那契数列	93
3.2.2	关系运算和逻辑运算	62	5.3 面向过程编程	94
3.2.3	选择结构的程序设计	64	5.3.1 跟我做：卡普雷卡尔 黑洞	94
3.2.4	动手练：判断三角形 构成	65	5.3.2 模块化程序设计	96
3.3 循环结构	66		5.3.3 动手练：数字黑洞 1 和 4	96
3.3.1	跟我做：冰雹猜想	67	第 6 章 调试程序	98
3.3.2	循环结构的程序设计	69	6.1 程序错误概述	98
3.3.3	动手练：肖像在哪里	71	6.1.1 产生程序错误的	

原因	98	8.4 角色特效	135
6.1.2 减少程序错误的方法	99	8.4.1 海底探险 5: 角色特效	135
6.2 程序调试方法	100	8.4.2 设置角色特效	136
6.2.1 使用数据显示器	100	8.4.3 动手练: 魔术师	137
6.2.2 输出日志	100	第 9 章 侦测	139
6.2.3 降低程序运行速度	101	9.1 距离侦测	139
6.2.4 设置断点	102	9.1.1 海底探险 6: 距离侦测	139
第 2 部分 图形编程篇		9.1.2 距离侦测积木	140
第 7 章 舞台	107	9.1.3 动手练: Pico 和小狗散步	140
7.1 海底探险 1: 游戏场景	107	9.2 碰撞侦测	142
7.1.1 游戏情节介绍	107	9.2.1 海底探险 7: 碰撞侦测	142
7.1.2 制作游戏场景	108	9.2.2 碰撞侦测积木	144
7.2 舞台布局和管理	110	9.2.3 动手练: 巡线甲虫	147
7.2.1 舞台布局和显示模式	110	9.3 键盘和鼠标侦测	148
7.2.2 舞台背景管理	113	9.3.1 海底探险 8: 扔炸弹	148
7.3 动手练: 电子相册	117	9.3.2 键盘和鼠标侦测积木	149
第 8 章 角色外观和运动	119	9.3.3 动手练: 反弹球	150
8.1 创建角色	119	9.4 时间侦测	151
8.1.1 海底探险 2: 创建角色	119	9.4.1 海底探险 9: 游戏倒计时	151
8.1.2 创建角色的方式	122	9.4.2 时间侦测积木	151
8.1.3 角色列表区	122	9.4.3 动手练: 时钟	153
8.2 角色造型	123	9.5 视频侦测	154
8.2.1 海底探险 3: 角色动画	123	9.5.1 跟我做: 人体感应开灯	154
8.2.2 造型控制	124	9.5.2 视频侦测积木	155
8.2.3 动手练: 街舞表演	125	9.5.3 动手练: 手势抓蝴蝶	157
8.3 角色运动	126	9.6 响度侦测	158
8.3.1 海底探险 4: 角色运动	126	9.6.1 跟我做: 声控开灯	158
8.3.2 运动控制	130	9.6.2 响度侦测积木	158
8.3.3 动手练: 飞行猫	134		

9.6.3	动手练：吹生日蜡烛	159
第 10 章	绘图	161
10.1	海底探险 10：游戏计分 和血条	161
10.2	画笔	164
10.2.1	跟我做：彩虹画板	164
10.2.2	画笔积木	165
10.2.3	动手练：颜色图谱	167
10.3	图章	167
10.3.1	跟我做：种蘑菇	167
10.3.2	图章积木	168
10.3.3	动手练：彩色风车	169
第 11 章	声音	171
11.1	海底探险 11：游戏音效	171
11.1.1	为游戏增加音效	171
11.1.2	复制更多角色	173
11.2	播放声音	174
11.3	乐器演奏	175
11.3.1	跟我做：模拟乐器	175
11.3.2	乐器演奏积木	177
11.3.3	弹奏鼓声积木	179
11.3.4	动手练：演奏 《小毛驴》	180
第 3 部分 进阶编程篇		
第 12 章	克隆	183
12.1	海底探险 12：克隆角色	183
12.1.1	克隆功能的使用	183
12.1.2	用克隆技术改造 游戏	185
12.2	面向对象编程	187
12.2.1	用角色模式编程	187
12.2.2	用克隆模式编程	188
12.3	动手练：飞扬的小鸟	190
第 13 章	消息和事件	192
13.1	海底探险 13：消息机制	192
13.1.1	广播和接收消息	192
13.1.2	用消息机制改造 游戏	193
13.2	消息机制	197
13.2.1	消息的异步和同步	197
13.2.2	消息队列的应用	197
13.3	事件驱动编程	199
13.3.1	事件的并发执行	199
13.3.2	避免事件的并发	200
13.4	动手练：小熊打车	201
第 14 章	英汉词典	202
14.1	搭建项目框架	202
14.2	词典排序功能	204
14.2.1	冒泡排序	204
14.2.2	选择排序	206
14.2.3	插入排序	207
14.2.4	快速排序	209
14.2.5	词典排序	211
14.3	词典查询功能	212
14.3.1	顺序查找	213

14.3.2	二分查找	214	15.2.2	迷宫数据结构	221
14.3.3	词典查询	216	15.2.3	加载迷宫地图	221
14.4	新增词条功能	216	15.2.4	构建迷宫外形	224
第 15 章	企鹅走迷宫	218	15.3	迷宫寻路	226
15.1	迷宫简介	218	15.3.1	算法策略简介	226
15.2	迷宫界面	219	15.3.2	迷宫算法分析	228
15.2.1	二维数组	220	15.3.3	迷宫算法实现	230

基础编程篇

Scratch 是一种简单易学的编程语言，对编程有兴趣的人都能很快学会使用。本书遵循由浅入深的原则编排内容，把编写基本的 Scratch 应用程序的内容安排在本书的第 1 部分，而涉及 Scratch 图形编程和进阶编程的内容安排在第 2、3 部分。本部分内容讲授基本的 Scratch 编程知识，采用鸡兔同笼、冰雹猜想、海伦公式等有趣且贴近中小学生学习数学知识的内容作为主要的编程案例，详细地向读者讲解变量、程序结构、过程等编程知识。通过学习基础编程篇，使初学者逐步建立起编程的知识体系和掌握编程的基本技能。

在基础编程篇中，我们先对 Scratch 开发环境和编写应用程序进行简单介绍，然后从编程的基本元素——“变量”开始讲授 Scratch 编程知识，接着讲授基本的算术运算、三角函数和随机数等数学运算的编程；之后讲授结构化程序设计、程序流程图、列表和过程等编程知识；最后讲授在 Scratch 中调试程序的一些常用方法。

完成基础编程篇的学习，读者将掌握编写应用程序的基本技能，中小学生学习将能够使用 Scratch 编程求解数学问题，把编程技能运用到数学学科的学习中。另外，经过编程入门教育之后，读者具备了基本的编程思想，就可以开始学习诸如 Python 和 C/C++ 等高级语言的编程了。

编程起步

欢迎走进 Scratch 编程的奇妙世界,从这里开始,本书将向没有编程基础的读者讲授如何使用 Scratch 编写应用程序。这一章将向初学者讲授学习本书其余部分内容需要掌握的一些基本概念和基础知识。

什么是编程?什么是 Scratch 编程?Scratch 积木式编程具有哪些优势?Scratch 编程语言的指令系统是怎样的?读者将在阅读本章的过程中找到这些问题的答案。同时,通过对本章的学习,读者将学会如何选择 Scratch 开发环境,学会安装 Scratch 离线编辑器和设置语言环境、字体大小等。之后,我们将手把手地教读者使用 Scratch 编辑器开发自己的应用程序项目,通过简单的临摹案例让读者快速熟悉 Scratch 开发环境和编程方法,为后续的学习作铺垫。此外,如果读者打算以后学习某种高级语言(如 Python、C/C++ 等),可参考本章给出的从 Scratch 到 Python 等高级语言的学习路径的建议进行学习。

本章包括以下主要内容。

- ◇ 介绍 Scratch 编程的特点、主要版本和项目构成等。
- ◇ 准备 Scratch 开发环境和设置编辑器的语言环境、字体大小等。
- ◇ 介绍 Scratch 编辑器界面的主要组成部分及其功能说明。
- ◇ 介绍 Scratch 指令系统和指令积木的特点、操作方法、功能类别等。
- ◇ 以临摹方式创作“韩信点兵”和“星际飞行”项目。

1.1 Scratch 编程概述

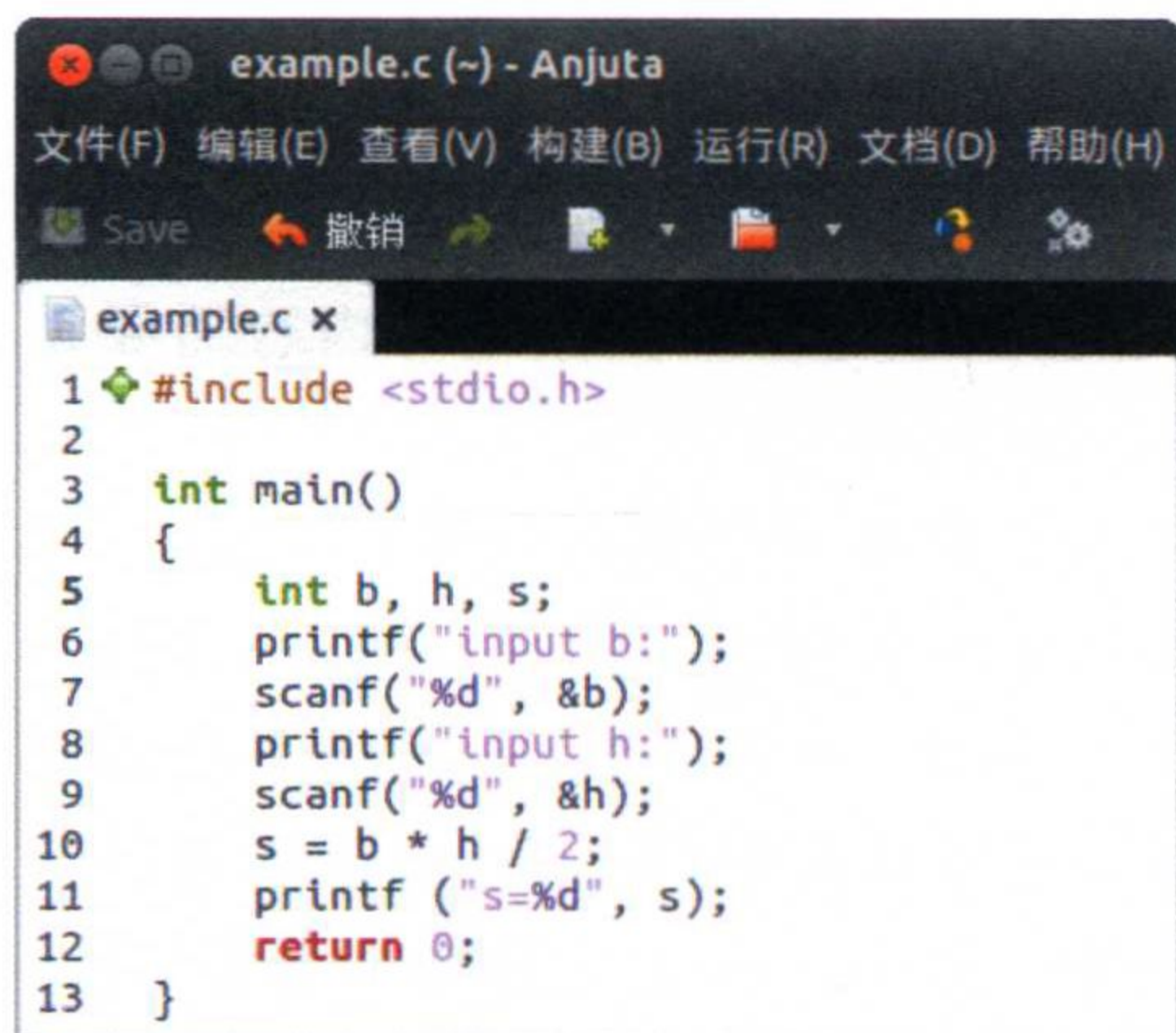
1.1.1 为什么用 Scratch 编程

在世界上第一台电子计算机 ENIAC 诞生后,各种编程语言陆续被计算机科学家创造出来。人们通过编程语言能够高效地与计算机系统进行交流,控制计算机按照人们的意愿进行工作。经过不断地发展和完善,一些编程语言与人类的自然语言和数学语言越来越接近,它们被称为高级语言,比如 C、C++、Java 和 Python 等就是全世界较为流行的高级语言。

为了利用计算机进行工作,人们使用某种编程语言将解决问题的方法和步骤描述成计算机能够理解和执行的一系列指令,这些指令的集合叫作计算机程序(简称程序,也叫作脚本),这个过程叫作编写程序(简称编程)。

如图 1-1-1 所示,这是一个使用 C 语言编写的计算三角形面积的示例程序。从中可以看到,C 语言使用英文字符来描述程序的指令(也叫作代码),其他高级语言也是如此。这是由于美国在早期计算机工业的发展中处于主要地位,因此一般的高级语言都是以英语为蓝本进行设计的。

我们还看到在这个 C 语言程序中,每一行代码都以一个分号结束。如果某一行代码末尾缺少一个分号,那么这个程序在编译时就会出现语法错误。如图 1-1-2 所示,这是在某个 IDE 环境中编译时因缺少分号而提示的错误信息。



```

example.c (-) - Anjuta
文件(F) 编辑(E) 查看(V) 构建(B) 运行(R) 文档(D) 帮助(H)
Save 撤销
example.c x
1 #include <stdio.h>
2
3 int main()
4 {
5     int b, h, s;
6     printf("input b:");
7     scanf("%d", &b);
8     printf("input h:");
9     scanf("%d", &h);
10    s = b * h / 2;
11    printf ("s=%d", s);
12    return 0;
13 }

```



```

Messages Build 37: noilinux x Build 35: noilinux x
Building in directory: /home/noilinux
make example.o
Compiling example.c --> example.o
example.c: In function 'main':
x example.c:10:2: error: expected ';' before 's'
s = b * h / 2;
^
make: *** [example.o] 错误 1
Completed unsuccessfully
Total time taken: 0 secs

```

图 1-1-1 计算三角形面积的 C 语言程序

图 1-1-2 编译 C 语言程序时缺少分号出现的错误信息

和 C 语言类似,其他高级语言也都有着严格的语法要求和各种编程规则。如果我们编写的程序没有遵守这些“金科玉律”,那么,轻则会收到错误或警告信息,重则可能导致计算机系统宕机。因此,在实际工作中进行编程需要经过专门的学习和训练,由专业程序员负责。

编程爱好者在学习 C 语言等编程语言时,一开始总是会遇到各种各样的麻烦和困难。尽管在计算机发展史上,很早就出现了诸如 BASIC 这种专门给普通编程爱好者使用的编程语言,并且在青少年中也有一定的使用量,但是这类编程语言仍然有着严格的语法要求和编程规则,将许多初学者的编程热情浇灭在起步阶段。

近年来,随着全球信息技术的飞速发展,专门为青少年开发的各种高级语言不断涌现,其中以 MIT Scratch 为代表的图形化编程语言从中脱颖而出,逐渐成为全球流行的、最适合对青少年进行编程教育的新一代编程语言。

如图 1-1-3 所示,这是使用 MIT Scratch 语言编写的计算三角形面积的两个程序脚本,它们分别是使用英文和中文进行描述的。这种使用 Scratch 编写的程序,由代表不同指令的积木块按照一定的逻辑关系组合而成。不需要背记复杂的语法规则,也不用担心程序会出现语法错误而无法运行,或者因为违反某些编程规则而导致程序崩溃。如此一来,就算是初学者也能将精力放在思考程序逻辑上,能够轻松地按照自己的意图选择不同功能的积木块,再将它们拼接组合成自己需要的程序。这种编写程序的方式充满了玩乐高积木般的乐趣。因此,Scratch 在 2007 年一经推出,就像一团熊熊烈火,迅速点燃了全球青少年的编程热情。

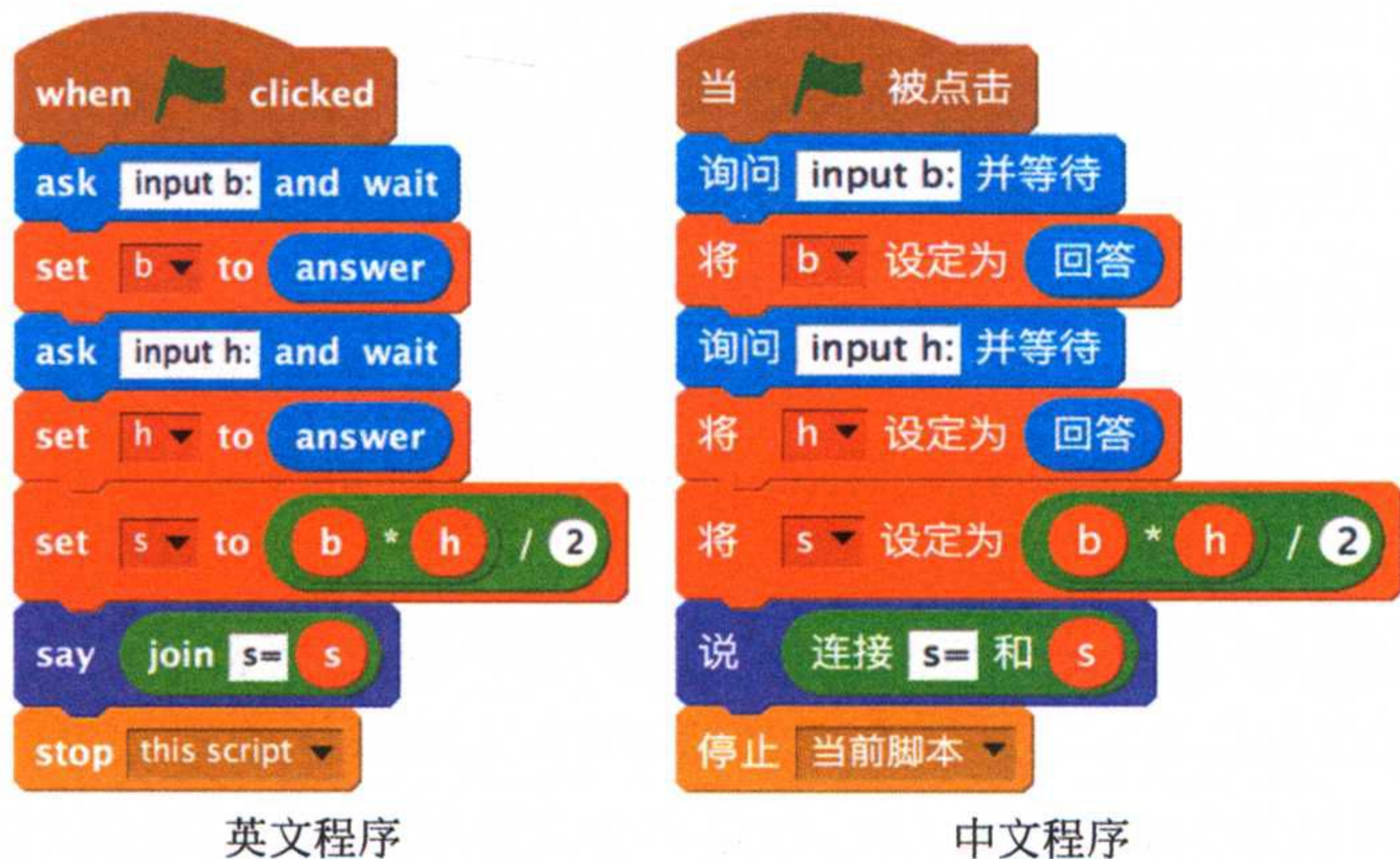


图 1-1-3 两个计算三角形面积的 Scratch 程序

这就是本书要向读者讲授的图形化编程语言——Scratch,它是由美国麻省理工学院(MIT)媒体实验室专门为8~16岁青少年设计开发的。使用这个具有魔力的Scratch作为编程工具,能够轻松地创作出各种交互式故事、游戏、动画、音乐、美术作品或其他应用程序,并通过Scratch社区将自己的创意作品分享给全世界的编程爱好者。

1.1.2 从 Scratch 到 Python

近年来,在中小学阶段推广和普及编程教育已成为全球各国的共识。2017年7月,国务院发文指出,将逐步在中小学阶段设置人工智能相关课程和推广编程教育。在全球流行编程教育的趋势之下,作为适合中小学生学习编程入门教育的图形化编程语言,Scratch的影响力日渐扩大,学习Scratch编程的青少年与日俱增。

Scratch是青少年编程教育的起点,而不是终点。为使Scratch能够被8~16岁的青少年学习和理解,Scratch开发团队刻意限制Scratch编程语言的功能和特性,保持Scratch简单易学的特色。因为Scratch的设计初衷是帮助青少年学习编程,而不是进行专业软件开发。

编程思想是灵魂,编程语言只是躯壳。当青少年通过Scratch掌握基本的编程思想之后,可以选择转向Python等具备完整编程特性的高级语言,就可以开发更为复杂的网络应用程序、数据库应用程序或人工智能应用程序,等等。

那么,从Scratch到Python应该如何前进呢?下面我们给出一个学习路径的建议供读者参考。

第1步:在中文界面下学习Scratch编程

在国内,8岁的儿童正处于小学低年级阶段,很多地方是从小学三年级起开设英语课。无论是汉字或英文,低龄儿童都存在认知上的困难,而Scratch图形化的特性有助于少年儿童学习编程。由于少儿编程教育在国内尚未普及,目前较为普遍的做法是家长通过教材自学编程之后再教自己的孩子。对多数人而言,在英文界面下进行Scratch编程有一定的困难,而Scratch对简体中文或其他众多语言的支持能把编程变得简单。如

图 1-1-3 中右边的程序是在简体中文界面下编写的,Scratch 指令积木块上的文字采用简体中文表示,对大多数家长来说更容易阅读和理解。同时,也可以减少为 Scratch 程序添加注释。因此,推荐初学者先在简体中文界面下学习 Scratch 编程。

第 2 步:在英文界面下学习 Scratch 编程

在简体中文界面下学习并掌握 Scratch 编程之后,就可以切换到英文界面下进行编程。当然,如果初学者有一定的英文基础,完全可以跳过第 1 步,直接使用全英文进行编程。

在英文界面下进行 Scratch 编程,是为以后学习 Python 等高级语言做准备。这是因为 Python 等高级语言的编程是基于文本的、使用英文关键字编写程序代码。所以,初学者在中文界面下掌握 Scratch 编程之后,作为一个过渡阶段,需要切换到英文界面去适应在英文环境中进行编程。

如图 1-1-4 所示,这是在 Scratch 英文界面下编写的插入排序算法的程序脚本。我们把它积木块的图形去除,只保留文字,将其转为文本型的脚本,之后再把它和使用 Python 语言编写的插入排序算法脚本进行对比,如图 1-1-5 所示。从文本的角度进行对比,可以看到 Scratch 和 Python 的程序是很相似的。因此,初学者先学习 Scratch 编程,在掌握基本的编程思想之后,再转向学习 Python 等高级语言编程,学习曲线会比较平缓。

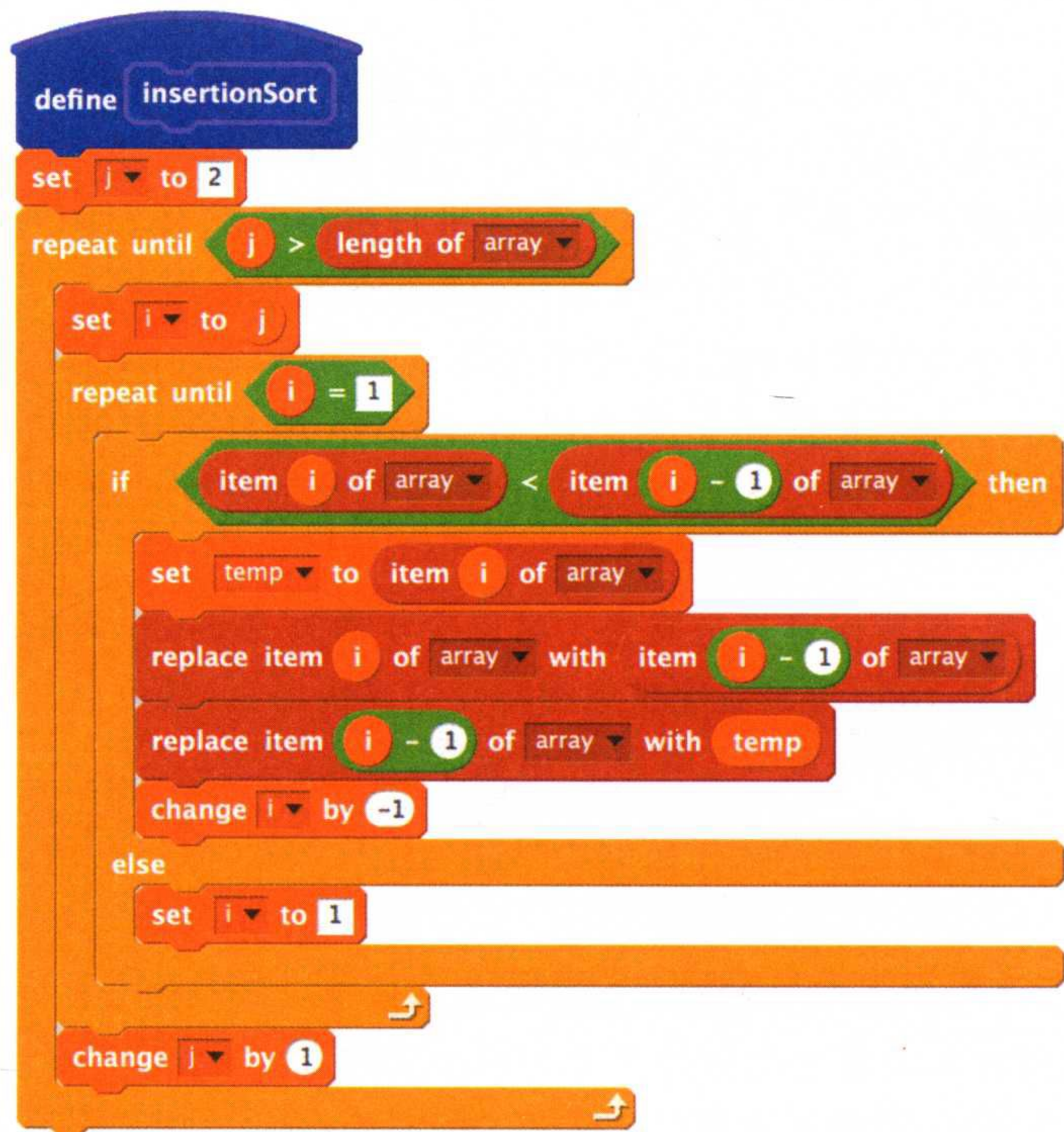


图 1-1-4 用 Scratch 编写的插入排序算法


```

1  define insertionSort
2    set j to 2
3    repeat until j > length of array
4      set i = j
5      repeat until i = 1
6        if item[i] of array < item[i-1] of array then
7          set temp to item[i] of array
8          replace item[i] of array with (item[i-1] of array)
9          replace item[i-1] of array with (temp)
10         change i by -1
11       else
12         set i to 1
13     change j by 1

```

Scratch脚本

```

1  def insertionSort():
2    j = 1
3    while j < len(array):
4      i = j
5      while i > 0:
6        if array[i] < array[i-1]:
7          temp = array[i]
8          array[i] = array[i-1]
9          array[i-1] = temp
10         i = i - 1
11       else:
12         i = 0
13     j = j + 1

```

Python脚本

图 1-1-5 用 Scratch 和 Python 编写的插入排序算法

第 3 步：学习 Python 或其他高级语言编程

在通过 Scratch 走上编程之路后，可以继续学习 Python、C/C++、Java 等高级语言。

在众多的高级语言中，Python 是一个不错的选择。它有着庞大的社区支持，各种技术资料非常丰富。同时，它有众多的编程库能够实现对各种新技术的支持，比如当下火热的人工智能和机器学习领域，都能找到相应的 Python 库。无论是客户端、云端，还是物联网终端，都能看到 Python 的身影，可以说，Python 的应用无处不在。总之，Python 的优点很多，在此不一一列举。

对 Python 或者其他高级语言编程知识的讲授已经超出本书的范畴，读者可以购买相关图书进行学习。

1.1.3 Scratch 的主要版本

2007 年 5 月，Scratch 的第一个版本发布，它基于 Squeak 平台和 Smalltalk 语言开发，可以运行于 Windows、Mac OS X 和 Debian/Ubuntu 等操作系统。目前仍然有不少用户使用的版本是 Scratch 1.4，如图 1-1-6 所示。如果你的计算机硬件配置较低或者使用的是 Windows XP 操作系统，则可以安装使用 Scratch 1.4 编辑器。

2013 年 5 月，Scratch 2.0 发布，它是基于 Adobe Flash 平台重新开发的，分为在线版和离线版。Scratch 2.0 是目前较为流行的版本，如图 1-1-7 所示。在支持 Adobe Flash Player 插件的浏览器（IE、Edge、Firefox、Safari 或 Chrome 等）中可以直接运行 Scratch 2.0 在线编辑器，不需要安装到用户的计算机上；而在安装有 Adobe AIR 运行环境的操作系统（Windows、Mac OS X 或 Linux）中，可以运行 Scratch 2.0 离线编辑器。在编写本书时，Scratch 2.0 的最新版本是 458.0.1。

和 Scratch 1.4 相比，Scratch 2.0 可谓是焕然一新，它新增的功能主要有：提供更为友好的图形用户界面；升级内置的声音编辑器和图像编辑器；通过支持矢量图形而改善舞台画面的质量；在线版提供的书包功能可以方便地管理常用的图片、声音等素材和脚本；增加视频侦测功能，可以实现手势控制等简单的体感技术应用。最为重要的是增加克隆功能和自定义过程功能，这能极大简化复杂应用程序的编写，提高代码的复用程度，使在 Scratch 2.0 中能够学习和应用面向对象和模块化的编程思想，有利于以后转向其