

LEARNING MADE EASY



3rd Edition

# Raspberry Pi<sup>®</sup>

for  
**dummies**<sup>®</sup>  
A Wiley Brand



Connect the Raspberry Pi  
and install the OS

Learn Programming with  
Scratch<sup>®</sup> and Python<sup>®</sup>

Create electronics projects  
connected to the GPIO port



Sean McManus  
Mike Cook





# Raspberry Pi<sup>®</sup>

3rd Edition

**by Sean McManus and Mike Cook**

for  
**dummies**<sup>®</sup>  
A Wiley Brand

## Raspberry Pi® For Dummies®, 3rd Edition

Published by: **John Wiley & Sons, Inc.**, 111 River Street, Hoboken, NJ 07030-5774, [www.wiley.com](http://www.wiley.com)

Copyright © 2017 by John Wiley & Sons, Inc., Hoboken, New Jersey

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the Publisher. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Trademarks:** Wiley, For Dummies, the Dummies Man logo, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and may not be used without written permission. Raspberry Pi is a registered trademark of Raspberry Pi Foundation. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002. For technical support, please visit <https://hub.wiley.com/community/support/dummies>.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit [www.wiley.com](http://www.wiley.com).

Library of Congress Control Number: 2017950656

ISBN: 978-1-119-41200-7; 978-1-119-41202-1 (ebk); 978-1-119-41201-4 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

# Contents at a Glance

<b>Introduction</b> .....	1
<b>Part 1: Setting Up Your Raspberry Pi</b> .....	7
CHAPTER 1: Introducing the Raspberry Pi .....	9
CHAPTER 2: Downloading the Operating System .....	23
CHAPTER 3: Connecting Your Raspberry Pi .....	35
<b>Part 2: Getting Started with Linux</b> .....	57
CHAPTER 4: Using the Desktop Environment .....	59
CHAPTER 5: Using the Linux Shell .....	87
<b>Part 3: Using the Raspberry Pi for Both Work and Play</b> ...	125
CHAPTER 6: Being Productive with the Raspberry Pi .....	127
CHAPTER 7: Editing Photos on the Raspberry Pi with GIMP .....	139
CHAPTER 8: Playing Audio and Video on the Raspberry Pi .....	149
<b>Part 4: Programming the Raspberry Pi</b> .....	161
CHAPTER 9: Introducing Programming with Scratch .....	163
CHAPTER 10: Programming an Arcade Game Using Scratch .....	183
CHAPTER 11: Writing Programs in Python .....	207
CHAPTER 12: Creating a Game with Python and Pygame Zero .....	239
CHAPTER 13: Programming Minecraft with Python .....	257
CHAPTER 14: Making Music with Sonic Pi .....	281
<b>Part 5: Exploring Electronics with the Raspberry Pi</b> .....	297
CHAPTER 15: Understanding Circuits .....	299
CHAPTER 16: Taking Control of Your Pi's Circuitry .....	321
CHAPTER 17: Lots of Multicolored LEDs .....	357
CHAPTER 18: Old McDonald's Farm and Other RFID Adventures .....	389
<b>Part 6: The Part of Tens</b> .....	425
CHAPTER 19: Ten Great Software Packages for the Raspberry Pi .....	427
CHAPTER 20: Ten Inspiring Projects for the Raspberry Pi .....	437
<b>Appendix A: Troubleshooting and Configuring the Raspberry Pi</b> .....	445
<b>Index</b> .....	459



# Table of Contents

<b>INTRODUCTION</b> .....	1
About This Book .....	1
Why You Need This Book .....	2
Foolish Assumptions .....	3
Icons Used in This Book .....	3
Beyond the Book .....	4
Where to Go from Here .....	4
<b>PART 1: SETTING UP YOUR RASPBERRY PI</b> .....	7
<b>CHAPTER 1: Introducing the Raspberry Pi</b> .....	9
Getting Familiar with the Raspberry Pi .....	11
Figuring Out What You Can Do with a Raspberry Pi .....	14
Getting Your Hands on a Raspberry Pi .....	15
Determining What Else You Need .....	16
<b>CHAPTER 2: Downloading the Operating System</b> .....	23
Introducing Linux .....	24
Creating a NOOBS Card .....	25
Downloading NOOBS .....	25
Formatting the SD card .....	26
Copying NOOBS to the SD or MicroSD card .....	30
Using Your NOOBS Card .....	32
Flashing an SD or MicroSD card .....	32
<b>CHAPTER 3: Connecting Your Raspberry Pi</b> .....	35
Inserting the SD Card .....	37
Connecting the Raspberry Pi Camera Module .....	39
Connecting the camera on a Pi Zero W .....	40
Connecting the camera on other Raspberry Pi models .....	40
Preparing Your Pi Zero or Zero W .....	41
Connecting a Monitor or TV .....	41
Connecting an HDMI or DVI display .....	42
Connecting a television using composite video .....	42
Connecting a USB Hub .....	43
Connecting a Keyboard and Mouse .....	44
Connecting Audio .....	44
Connecting to Your Router .....	45
Connecting the Power and Turning on the Raspberry Pi .....	46

Logging In . . . . .	50
Configuring Your Raspberry Pi in Raspbian. . . . .	50
Configuring Your Wi-Fi . . . . .	53
Configuring Bluetooth Devices . . . . .	54
Testing the Camera Module. . . . .	54
Setting Up the Data Partition. . . . .	56
Taking Your Next Steps with the Raspberry Pi . . . . .	56
<b>PART 2: GETTING STARTED WITH LINUX. . . . .</b>	<b>57</b>
<b>CHAPTER 4: Using the Desktop Environment. . . . .</b>	<b>59</b>
Navigating the Desktop Environment . . . . .	60
Using the Applications menu. . . . .	60
Running programs that are not on the menu. . . . .	63
Resizing and closing program windows. . . . .	63
Using the Task Manager . . . . .	64
Using File Manager . . . . .	65
Navigating File Manager . . . . .	66
Copying and moving files and folders . . . . .	69
Selecting multiple files and folders. . . . .	70
Creating new folders and blank files . . . . .	71
Deleting files and folders . . . . .	71
Changing how files are displayed . . . . .	72
Opening a folder in the terminal. . . . .	73
Browsing the Web . . . . .	74
Using Chromium to browse the web . . . . .	74
Searching within web pages . . . . .	76
Using tabbed browsing. . . . .	76
Adding and using bookmarks . . . . .	76
Protecting your privacy. . . . .	77
Sending and Receiving Email with Claws Mail. . . . .	78
Using the Image Viewer . . . . .	79
Using the Text Editor. . . . .	81
Customizing the Desktop. . . . .	82
Finding and Installing New Applications . . . . .	83
Backing Up Your Data . . . . .	84
Logging Out from PIXEL and Shutting Down. . . . .	85
<b>CHAPTER 5: Using the Linux Shell. . . . .</b>	<b>87</b>
Understanding the Prompt . . . . .	88
Exploring Your Linux System . . . . .	88
Listing files and directories . . . . .	88
Changing directories. . . . .	89
Checking file types. . . . .	89



Changing to the parent directory .....	90
Understanding the directory tree .....	91
Using relative and absolute paths .....	94
Investigating more advanced listing options .....	96
Understanding the Long Listing Format and Permissions .....	99
Slowing Down the Listing and Reading Files with the Less Command .....	102
Speeding Up Entering Commands .....	103
Using Redirection to Create Files .....	104
Creating Directories .....	106
Deleting Files .....	107
Using Wildcards to Select Multiple Files .....	108
Removing Directories .....	110
Copying and Renaming Files .....	111
Installing and Managing Software on Your Raspberry Pi .....	113
Updating the cache .....	114
Finding the package name .....	114
Installing software .....	115
Running software .....	116
Upgrading the software .....	116
Removing software and freeing up space .....	117
Finding out what's installed .....	118
Managing User Accounts on Your Raspberry Pi .....	118
Learning More about Linux Commands .....	120
Customizing the Shell with Your Own Linux Commands .....	122
Shutting Down and Rebooting Your Raspberry Pi .....	124

## **PART 3: USING THE RASPBERRY PI FOR BOTH WORK AND PLAY** .....

<b>CHAPTER 6: Being Productive with the Raspberry Pi</b> .....	127
Installing LibreOffice on Your Raspberry Pi .....	128
Working with LibreOffice on the Raspberry Pi .....	128
Saving your work .....	129
Writing letters in LibreOffice Writer .....	129
Managing your budget in LibreOffice Calc .....	131
Creating presentations in LibreOffice Impress .....	134
Creating a party invitation with LibreOffice Draw .....	136
<b>CHAPTER 7: Editing Photos on the Raspberry Pi with GIMP</b> .....	139
Working with GIMP .....	140
Understanding the GIMP screen layout .....	140
Resizing an image in GIMP .....	142
Cropping your photo .....	144

	Rotating and flipping your photo .....	145
	Adjusting the colors.....	145
	Fixing imperfections .....	146
	Converting images between different formats.....	147
	Finding Out More about GIMP.....	147
<b>CHAPTER 8:</b>	<b>Playing Audio and Video on the Raspberry Pi .....</b>	<b>149</b>
	Setting Up Your Media Center.....	149
	Navigating the Media Center.....	150
	Adding Media.....	151
	Adding music .....	152
	Adding videos.....	153
	Adding pictures .....	154
	Streaming media .....	155
	Playing Music .....	155
	Playing Videos .....	156
	Viewing Photos .....	157
	Changing the Settings.....	157
	Using a Remote Control .....	158
	Turning Off Your Media Center .....	158
	Playing Music in the Desktop Environment.....	158
	<b>PART 4: PROGRAMMING THE RASPBERRY PI .....</b>	<b>161</b>
<b>CHAPTER 9:</b>	<b>Introducing Programming with Scratch.....</b>	<b>163</b>
	Understanding What Programming Is.....	164
	Working with Scratch .....	164
	Understanding the Scratch screen layout .....	165
	Positioning and resizing your sprite.....	166
	Making your sprite move .....	167
	Changing your sprite's appearance .....	172
	Adding sounds and music .....	176
	Creating scripts .....	178
	Using the Wait block to slow down your sprite.....	180
	Saving your work.....	180
	What's new in Scratch 2 .....	181
<b>CHAPTER 10:</b>	<b>Programming an Arcade Game Using Scratch .....</b>	<b>183</b>
	Starting a New Scratch Project and Deleting Sprites .....	184
	Changing the Background .....	185
	Adding Sprites to Your Game .....	185
	Drawing Sprites in Scratch.....	186
	Naming Your Sprites.....	189

Controlling When Scripts Run . . . . .	190
Using the green flag to start scripts . . . . .	190
Using the Forever Control block . . . . .	191
Enabling keyboard control of a sprite . . . . .	192
Enabling a sprite to control another sprite . . . . .	193
Using Random Numbers . . . . .	195
Detecting When a Sprite Hits Another Sprite . . . . .	196
Introducing Variables . . . . .	197
Making Sprites Move Automatically . . . . .	199
Fixing the Final Bug . . . . .	201
Adding Scripts to the Stage . . . . .	203
Duplicating Sprites . . . . .	203
Playing Your Game . . . . .	204
Adapting the Game's Speed . . . . .	204
Taking It Further with Scratch . . . . .	205
<b>CHAPTER 11: Writing Programs in Python . . . . .</b>	<b>207</b>
Working with Python . . . . .	208
Entering your first Python commands . . . . .	208
Using the shell to calculate sums . . . . .	210
Creating the Times Tables Program . . . . .	212
Creating and running your first Python program . . . . .	213
Using variables . . . . .	214
Accepting user input . . . . .	216
Printing words, variables, and numbers together . . . . .	216
Using for loops to repeat . . . . .	218
Creating the Chatbot Program . . . . .	221
Introducing lists . . . . .	222
Using lists to make a random chat program . . . . .	225
Adding a while loop . . . . .	227
Using a loop to force a reply from the user . . . . .	228
Using dictionaries . . . . .	229
Creating your own functions . . . . .	231
Creating the dictionary look-up function . . . . .	234
Creating the main conversation loop . . . . .	236
Final thoughts on Chatbot . . . . .	236
The final Chatbot program . . . . .	237
<b>CHAPTER 12: Creating a Game with Python and Pygame Zero . . . . .</b>	<b>239</b>
Collecting Your Sounds and Images . . . . .	240
Setting Up Your Folders . . . . .	241

Creating and Running Your First Program.....	241
Detecting mouse clicks.....	244
Animating your actors.....	245
Using random numbers.....	247
Adding more clouds.....	248
Making the clouds regenerate.....	250
Enabling multiple clouds to be clicked.....	251
Adding the timer.....	253
Adjusting the game difficulty.....	253
The final game listing.....	254
Exploring Pygame Zero Further.....	256
<b>CHAPTER 13: Programming Minecraft with Python.....</b>	<b>257</b>
Playing Minecraft.....	258
Moving around.....	258
Making and breaking things.....	259
Preparing for Python.....	260
Using the Minecraft Module.....	261
Understanding coordinates in Minecraft.....	261
Repositioning the player.....	262
Adding blocks.....	263
Stopping the player from changing the world.....	264
Setting the maze parameters.....	265
Laying the foundations.....	266
Placing the maze walls.....	267
Understanding the maze algorithm.....	268
Setting up the variables and lists.....	269
Creating the functions.....	270
Creating the main loop.....	271
Adding a ceiling.....	274
Positioning the player.....	274
The final code.....	275
Adapting the Program.....	279
<b>CHAPTER 14: Making Music with Sonic Pi.....</b>	<b>281</b>
Understanding the Sonic Pi Screen Layout.....	282
Playing Your First Notes.....	283
Using Note and Chord Names.....	285
Writing Shorter Programs.....	286
Composing Random Tunes Using Shuffle.....	287
Changing the Random Number Seed.....	287
Using List Names in Your Programs.....	288
Playing Random Notes.....	288

Experimenting with Live Loops . . . . .	289
Using Samples . . . . .	291
Adding Special Effects . . . . .	292
Synchronizing with Your Drumbeat . . . . .	293
Bringing It All Together . . . . .	293
Next Steps with Sonic Pi . . . . .	295

## **PART 5: EXPLORING ELECTRONICS WITH THE RASPBERRY PI . . . . .**

297

### **CHAPTER 15: Understanding Circuits . . . . .**

299

Discovering What a Circuit Is . . . . .	300
Understanding the nature of electricity . . . . .	300
Determining how a component needs to be treated . . . . .	307
Getting Familiar with the GPIO . . . . .	308
Putting the general purpose in GPIO . . . . .	309
Understanding what GPIOs do . . . . .	309
Putting an output pin to practical use . . . . .	311
Using GPIOs as inputs . . . . .	313
Learning which end is hot: Getting to grips with a soldering iron . . . . .	314
Making a soldered joint . . . . .	316
Looking at Ready-Made Add-On Boards . . . . .	317
The Sense HAT . . . . .	318
The Skywriter HAT . . . . .	318
The Xtrinsic Sense board . . . . .	319
Other boards . . . . .	319

### **CHAPTER 16: Taking Control of Your Pi's Circuitry . . . . .**

321

Accessing Raspberry Pi's GPIO Pins . . . . .	321
Soldering the GPIO pins onto Pi Zero . . . . .	323
Getting at all the pins with one connector . . . . .	324
Connecting things together . . . . .	326
Your First Circuit . . . . .	327
Bringing your LED to life . . . . .	328
Using Scratch 1.4 . . . . .	328
Control the flashing speed with an input . . . . .	329
Using Python . . . . .	331
Using GPIO ZERO . . . . .	334
Starting Out with a Dice Display . . . . .	337
A dice display . . . . .	337
The project . . . . .	340
The numbers . . . . .	340
The display . . . . .	341
Taking it further . . . . .	346

Pedestrian Crossing .....	347
The Pedestrian Crossing hardware.....	351
The Pedestrian Crossing software .....	352
Taking it further.....	356
<b>CHAPTER 17: Lots of Multicolored LEDs .....</b>	<b>357</b>
Making Colors .....	359
Using diffusers.....	359
Making more colors .....	360
The Way Forward.....	362
Bit-banging the APA102C protocol .....	365
Creating a class .....	366
Rainbow Invaders .....	370
Keepy Uppy .....	375
LEDs Galore .....	378
Current limits .....	379
Signals and memory .....	379
Display update .....	381
Getting more LEDs .....	381
<b>CHAPTER 18: Old McDonald’s Farm and Other RFID Adventures .....</b>	<b>389</b>
How RFID Work .....	390
A MIFARE card’s structure .....	393
A simple RFID jukebox .....	395
A better RFID jukebox.....	397
Taking it further.....	401
Dressing Up a Paper Doll .....	401
Runway time.....	406
Old McDonald’s Farm .....	410
Making sound samples.....	411
Making the graphics .....	413
<b>PART 6: THE PART OF TENS.....</b>	<b>425</b>
<b>CHAPTER 19: Ten Great Software Packages for the Raspberry Pi .....</b>	<b>427</b>
Penguins Puzzle.....	428
FocusWriter .....	429
Mathematica .....	429
XInvaders 3D .....	431
Fraqtive.....	431
Tux Paint.....	432
Grisbi.....	433

	Beneath a Steel Sky.....	434
	Sense HAT Emulator.....	435
	Brain Party.....	436
<b>CHAPTER 20:</b>	<b>Ten Inspiring Projects for the Raspberry Pi.....</b>	<b>437</b>
	One-Button Audiobook Player.....	437
	Weather Station.....	438
	Heart Rate Monitor.....	439
	Electric Skateboard.....	439
	T-Shirt Cannon.....	439
	Panflute Hero.....	440
	Magic Mirror.....	440
	Pi in the Sky.....	441
	Raspberry Turk.....	442
	Sound Fighter.....	443
	<b>APPENDIX A: TROUBLESHOOTING AND CONFIGURING THE RASPBERRY PI.....</b>	<b>445</b>
	<b>INDEX.....</b>	<b>459</b>





# Introduction

The Raspberry Pi is at the forefront of the maker movement, where people make their own inventions using a mixture of traditional craft skills and modern coding and electronics knowledge. It's also given more and more people access to a computer that provides a gateway into programming, electronics, and the world of Linux — the technically powerful (and free) rival to Windows and Mac OS. As a supercheap computer, the Raspberry Pi is also being pressed into service in media centers and as a family computer for games, music, photo editing, and word processing.

Although the Raspberry Pi presents new opportunities to everyone, it can also be a daunting prospect. It comes as a bare circuit board, so to do anything with it, you need to add an operating system on an SD or microSD card and connect it up to a screen, mouse, and keyboard. To get started, you need to learn a few basics of Linux, or at least get acquainted with PIXEL, the graphical desktop. You might be a geek who relishes learning new technologies, or you might be someone who wants a new family computer to use with the children. In either case, *Raspberry Pi For Dummies*, 3rd Edition, helps you get started with your Raspberry Pi and teaches you about some of the many fun and inspiring things you can do with it.

## About This Book

*Raspberry Pi For Dummies*, 3rd Edition, provides a concise and clear introduction to the terminology, technology, and techniques that you need to get the most from your Pi. With this book as your guide, you'll learn how to

- » Connect your Raspberry Pi.
- » Change its settings so that it works optimally for you.
- » Discover and install great free software you can use on your Raspberry Pi.
- » Use the desktop environment to run programs, manage files, surf the web, and view photos.
- » Use the Linux command line to manage your Raspberry Pi and its files.
- » Use the Raspberry Pi as a productivity tool.

- » Edit photos.
- » Play music and video.
- » Create animations and arcade games with the child-friendly Scratch programming language.
- » Write your own games and other programs using the Python programming language.
- » Compose music by programming with Sonic Pi.
- » Get started with electronics, from an introduction to soldering to the design and creation of electronic projects controlled by the Raspberry Pi.

Incidentally, within this book, you may note that some web addresses break across two lines of text. If you're reading this book in print and want to visit one of these web pages, simply key in the web address exactly as it's noted in the text, pretending as though the line break doesn't exist. If you're reading this as an ebook, you've got it easy — just click or tap the web address to be taken directly to the web page.

## Why You Need This Book

After you shake the Raspberry Pi out of the little electrostatic bag it comes in, what next?

This book answers that question. It enables you to get your Raspberry Pi up and running and also introduces you to some of the great things you can do with it, through satisfying practical projects. With this book as your companion, you can write games and other programs and create your own electronic gadgets, all with no prior programming knowledge.

The Raspberry Pi is most likely a bit different compared to other computers you've used, so this book also helps you to do some of the things on your Pi that you expect of every computer, such as play music and edit documents.

You can learn a lot of this through trial and error, of course, but that can be a frustrating way to spend your time. Using this book as a reference, you can more quickly start using your Raspberry Pi, whatever you plan to do with it.

# Foolish Assumptions

*Raspberry Pi For Dummies*, 3rd Edition, is written for beginners, by which we mean people who have never used a similar computer. However, we do have to make a few assumptions in writing this book, because we wouldn't have enough space for all its cool projects if we had to start by explaining what a mouse is! Here are our assumptions:

- » **You are familiar with other computers, such as Windows or Apple computers.** In particular, we assume that you're familiar with using windows, icons, and the keyboard and mouse, and that you know the basics of using your computer for things like browsing the Internet, writing letters, or copying files.
- » **The Raspberry Pi is not your only computer.** At times, you'll need to have access to another computer — for example, to create your SD or microSD card for the Pi. (See Chapter 2.) When it comes to networking, we assume you already have a router set up with an Internet connection and a spare port that you can plug the Raspberry Pi into.
- » **The Raspberry Pi is your first Linux-based computer.** If you're a Linux ninja, this book still gives you a solid reference on the Raspberry Pi and the version of Linux it uses, but no prior Linux knowledge is required.
- » **You share our excitement.** The Raspberry Pi can open up a world of possibilities to you!

Other than these assumptions, we hope this book is approachable for everyone. The Raspberry Pi is being adopted in classrooms and youth groups, and this book is a useful resource for teachers and students. The Raspberry Pi is also finding its way into many homes, where people of all ages (from children to adults) are using it for education and entertainment.

## Icons Used in This Book

If you've read other *For Dummies* books, you know that they use icons in the margin to call attention to particularly important or useful ideas in the text. In this book, we use four such icons:



TIP

The Tip icon highlights expert shortcuts or simple ideas that can make life easier for you.



TECHNICAL  
STUFF

Arguably, the whole book talks about technical stuff, but this icon highlights something that's *particularly* technical. We've tried to avoid unnecessary jargon and complexity, but some background information can give you a better understanding of what you're doing, and sometimes we do need to get quite techy, given the sophistication of the projects you're doing. Paragraphs highlighted with this icon might be worth rereading, to make sure you understand, or you might decide that you don't need to know that much detail. It's up to you!



REMEMBER

Although we'd like to think that reading this book is an unforgettable experience, we've highlighted some points that you might want to particularly commit to memory. They're either important takeaways, or they're fundamental to the project you're working on.



WARNING

As you would do on the road, slow down when you see a Warning icon. It highlights an area where things could go wrong.

## Beyond the Book

In addition to what you're reading right now, this book comes with a free access-anywhere Cheat Sheet with tips on installing software and using Scratch. To get this Cheat Sheet, simply go to [www.dummies.com](http://www.dummies.com) and type **Raspberry Pi Dummies Cheat Sheet** in the Search box.

Also be sure to check out this book's companion website ([www.dummies.com/extras/raspberrypi3e](http://www.dummies.com/extras/raspberrypi3e)), where you can download the code listings that appear throughout this book, as well as two bonus appendices (one on The GPIO on the Raspberry Pi and one on the RISC OS) as well as a bonus chapter on Mathematica, a mathematical program.

Both of us maintain our own personal websites too, which contain some additional information on the Raspberry Pi. Mike's is at [www.thebox.myzen.co.uk/Raspberry/Punnet.html](http://www.thebox.myzen.co.uk/Raspberry/Punnet.html), and Sean's is at [www.sean.co.uk](http://www.sean.co.uk).

## Where to Go from Here

It's up to you how you read this book. It's been organized to take you on a journey from acquiring and setting up your Raspberry Pi to learning the software that comes with it, and from writing your own programs to finally creating your own

electronics projects. Some chapters build on knowledge gained in earlier chapters, especially the sections on Scratch and Python — and all of Part 5.

We understand, though, that some projects or topics might interest you more than others, and you might need help in some areas right now. When a chapter assumes knowledge from elsewhere, we include cross-references to help you quickly find what you might have missed. We also include some signposts to future chapters, so you can skip ahead to a later chapter if it provides the quickest answer for you.

If you haven't set up your Pi yet, start with Part 1. If you have your Pi up and running, Part 2 shows you how to use the software on it. Part 3 covers productivity, creativity, and entertainment software. To flex your programming muscles, perhaps for the first time, read Part 4. You can learn Scratch, Python, or Sonic Pi here, and feel free to start with any one of those languages. The Python chapters provide a good foundation for Part 5, where you can start building your own electronics projects.



# 1

## Setting Up Your Raspberry Pi

## **IN THIS PART . . .**

Get to know the Raspberry Pi and what other equipment you will need to be able to use it.

Download the Linux operating system and prepare an SD or MicroSD card for use on your Raspberry Pi.

Connect your Raspberry Pi to the power, USB hub, keyboard, mouse, and screen.

Install and test the Raspberry Pi Camera Module.

Change the settings on your Raspberry Pi.



- » Getting up close and personal with the Raspberry Pi
- » Taking stock of your Raspberry Pi
- » Determining its limitations
- » Purchasing your very own Raspberry Pi
- » Figuring out what else you need

## Chapter **1**

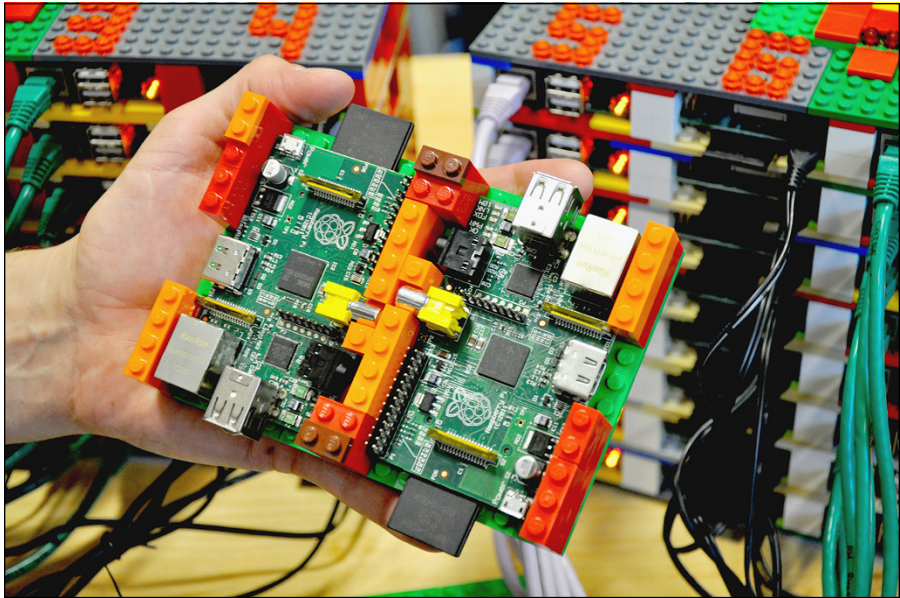
# Introducing the Raspberry Pi

**T**he Raspberry Pi is perhaps the most inspiring computer available today. Although most of the computing devices being used (including phones, tablets, and game consoles) are designed to stop people from tinkering with them, the Raspberry Pi is exactly the opposite. From the moment you see its shiny green circuit board, it invites you to prod it, play with it, and create with it. It comes with the tools you need to start creating your own software (or *programming*), and you can connect your own electronic inventions to it. It's cheap enough that if you break it, it won't break the bank, so you can experiment with confidence.

Lots of people are fired up about its potential, and they're discovering exciting new ways to use it. Dave Akerman ([www.daveakerman.com](http://www.daveakerman.com)) and friends attached one to a weather balloon and sent it nearly 40 kilometers high to take pictures of the Earth from near space using a webcam. (You can read about Dave's ballooning project in Chapter 20.)

Professor Simon Cox and his team at the University of Southampton connected 64 Raspberry Pi boards to build an experimental supercomputer, held together by Lego bricks. In the supercomputer (see Figure 1-1), the Raspberry Pis work together to solve a single problem. The project has been able to cut the cost of a

supercomputer from millions of dollars to thousands or even hundreds of dollars, making supercomputing much more accessible to schools and students. Others have also experimented with combining the processing power of multiple Pis. There's even an off-the-shelf kit you can use to combine four Raspberry Pi Zeros with a full-size Raspberry Pi (the Cluster HAT from Pimoroni) so that you can experiment with running programs across multiple Pis at the same time.



**FIGURE 1-1:**  
Two of the Raspberry Pi boards used in the University of Southampton's supercomputer, with the rest of the supercomputer in the background.

*Courtesy of Simon Cox and Glenn Harris, University of Southampton.*

The Pi is also being used to make weather stations, fitness gadgets, gaming devices, audiobook players, electric skateboards, and much more, as you discover in Chapter 20.

Although those projects are grabbing headlines, another story is less visible but more important: the thousands of people of all ages who are taking their first steps in computer science, thanks to the Raspberry Pi.

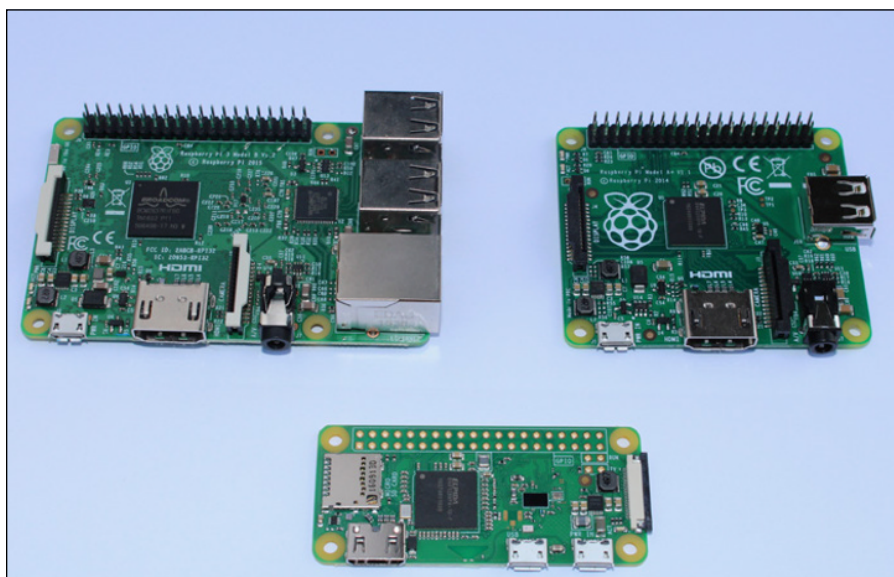
Both of the authors of this book used computers in the 1980s, when the notion of a home computer first became a reality. Back then, computers were less friendly than they are today. When you switched them on, you were faced with a flashing cursor and had to type something in to get it to do anything. As a result, though, a whole generation grew up knowing at least a little bit about how to give the computer commands, and how to create programs for it. As computers started to use mice and windows, people didn't need those skills any more, and they lost touch with them.

Eben Upton, designer of the Raspberry Pi, noticed the slide in skill levels when he was working at Cambridge University’s computer laboratory in 2006. Students applying to study computer science started to have less experience with programming than students of the past did. Upton and his university colleagues hatched the idea of creating a computer that would come supplied with all the tools needed to program it — and would sell for a target price of \$25 (about £20). It had to be able to do other interesting things, too, so that people were drawn to use it, and it had to be robust enough to survive being pushed in and out of school bags hundreds of times.

That idea started a six-year journey that led to the Raspberry Pi you probably have on your desk now as you read this book. It was released in February 2012, and sold half a million units by the end of the quarter. By July 2017, there were more than 14 million Raspberry Pis in homes, schools and workplaces, 10 million of them made in the UK. It is, by a large margin, the best-selling British computer of all time.

## Getting Familiar with the Raspberry Pi

When your Raspberry Pi arrives, you’ll see that it’s a circuit board, with components and sockets stuck on it, as shown in Figure 1-2. In an age when most computing devices are sleek and shiny boxes, the spiky Pi, with tiny codes printed in white all over it, seems alien. That’s a big part of its appeal, though: Many of the cases you can buy for the Raspberry Pi are transparent because people love the look of it.



**FIGURE 1-2:**  
The Raspberry Pi 3 Model B (top left), Model A+ (top right), and Pi Zero W (bottom).

Over the years, the Raspberry Pi has evolved, increasing its memory, improving its performance, and adding features. So which one should you get? Here's an overview designed to help you decide:

» **Raspberry Pi 3 Model B:** The third generation of the Raspberry Pi, it represents the best all-round Raspberry Pi at the time of writing. According to the Raspberry Pi Foundation, it is 50 to 60 percent faster than the previous model, and ten times faster than the original Raspberry Pi. It includes 1GB of memory, four USB ports, built-in Wi-Fi and Bluetooth, and an Ethernet port for a wired Internet connection. It has 40 General Purpose Input/Output (GPIO) pins, which you can use to connect to your own electronics projects. Like previous Pi models, it's about the size of a credit card. As with any current Raspberry Pi, it uses a MicroSD card for storage. If you're not sure which model to get and your budget allows, get this one. It represents the fastest performance, and offers the best experience on the desktop. Its price is around \$35 (about £30).

It's called the Model B, incidentally, as a tribute to the BBC Microcomputer that was popular in the U.K. in the 1980s. It's sobering to think that the BBC Micro cost about ten times the price of a Raspberry Pi, which, thanks to 35 years of progress in computer science, has more than 7,800 times more memory.

» **Raspberry Pi 1 Model A+:** A cut-down model, it is ideal for projects that need lower power consumption — typically battery-based projects. It is suitable for robots and projects in remote locations, where a wired electricity supply isn't viable and batteries must be used instead. It does not have an Ethernet socket, and only has one USB port, although you can connect it to a USB hub to use more devices simultaneously. It does have the full complement of 40 GPIO pins, though, so you should find that your projects and add-ons work with it. Like the Model B, it includes an audio output (headphones-style) socket. This model has 512MB of memory and a price of \$20 (or £20). The Model A+ is slightly shorter on the long side than the Raspberry Pi 3, measuring 6.5 centimeters by 5.5 centimeters.

» **Raspberry Pi Zero:** The Raspberry Pi Foundation astounded everyone when it gave this computer away with the print edition of its magazine *The MagPi*. The Raspberry Pi Zero measures 6.5 centimeters by 3 centimeters, is extremely lightweight, and has 512MB of memory and one Micro USB port. If you want to use the GPIO, you'll need to solder or affix your own pins, available separately. (You can read about adding GPIO pins to the Pi Zero in Chapter 16.) You'll also need a converter for the Mini HDMI socket, and for the Micro USB socket, so you should expect to spend a bit more than the price of the Pi (and have a bit more complexity in your setup). Billed as the \$5 computer, the Raspberry Pi Zero has at times been difficult to get hold of, which is perhaps not surprising given the phenomenal demand for it.

» **Raspberry Pi Zero W:** Released in February 2017, the Raspberry Pi Zero W added Wi-Fi and Bluetooth, and compatibility with the Raspberry Pi Camera Module. The Pi Zero W costs around \$10 (or about £10). If you're happy to solder your own GPIO pins, or you don't need them, the Raspberry Pi Zero or Zero W represents a great entry point to the Raspberry Pi family. After the Raspberry Pi Model 3, the Pi Zero W is our recommended best buy.

Of course, the older Raspberry Pis are still out there. Recent models usually remain in production while there is demand, and you can buy secondhand versions online from websites such as eBay. Generally speaking, the newer the model, the faster its performance. Memory upgrades have made a difference, as well as the use of more powerful processors, as the Pi has evolved. There are plenty of uses for the Pi that don't need especially fast performance, though, so you might find that an older Pi is perfect for your project. The older models are described in this list:

» **Raspberry Pi 1 Model B with 256MB memory:** Although it's called Model B, this was the first Raspberry Pi to be released, in February 2012. The Raspberry Pi Model B features an Ethernet connection for the Internet and two USB ports. It uses an SD card for storage.

» **Raspberry Pi 1 Model B with 512MB memory:** Released in October 2012, the Raspberry Pi Model B had twice the memory capacity. This improved the speed of some software, especially applications that used images heavily.

» **Raspberry Pi 1 Model A:** The Model A, released in February 2013, is a stripped-down version of the Model B. It has just one USB port and doesn't have an Ethernet port for connecting to the Internet. It has 256MB of memory.

» **Raspberry Pi 1 Model B+:** The Model B+, released in July 2014, has been described by the Raspberry Pi Foundation as "the final evolution of the original Raspberry Pi." It runs all the same software as the previous versions of the Raspberry Pi, but it has four USB ports, more GPIO pins for connecting electronics projects to the Pi, and lower power consumption and better audio than the Model B. In common with the Model B, it has 512MB of memory. Although all previous versions use SD cards for data storage, the Model B+ introduced the smaller MicroSD cards, which are now standard on the Raspberry Pi.

» **Raspberry Pi 2 Model B:** Launched in February 2015, this model doubled the memory on the Model B+ to 1GB. It increased performance, compared to the Model B+, while retaining its physical features. Over the years the Pi's performance has been improved through new software releases as well as updates to the hardware. The Pi 2 represents an immediately noticeable speed-up, compared to the Model B+.



REMEMBER

If you're using anything earlier than the Model B+, you'll need full-size SD cards (not MicroSD) for storage, and you'll only have 26 GPIO pins to play with. Current add-ons are unlikely to be compatible with the early boards, so check their requirements before you buy.

In this book, we offer guidance on older models where possible but will assume you're using at least a Model B+ for the projects. For best performance, we recommend using a current model, if possible.

You'll also see the Raspberry Pi Compute Module in the online stores alongside the Raspberry Pi, but this is something quite different. It's aimed at engineers creating industrial applications (known as embedded systems) or products based on Raspberry Pi technology. We only mention it here in case you wonder what it is: It's not covered further in this book, and it's almost certainly not what you want to buy for your first Raspberry Pi.



TECHNICAL  
STUFF

The Raspberry Pi was made possible in part by the advances in mobile computer chips that have happened in recent years. At its heart is a Broadcom chip (BCM2835, BCM2836, or BCM2837) that contains an ARM central processing unit (CPU) and a Videocore IV graphics processing unit (GPU). The CPU and GPU share the memory between them. The GPU is powerful enough to be able to handle Blu-ray quality video playback.

Instead of running Windows or Mac OS, the Raspberry Pi uses an operating system called Linux. It's a leading example of open source, a completely different philosophy to the commercial software industry. Rather than being created within the heavily guarded walls of a company, with its design treated as a trade secret, Linux is built by companies and expert volunteers working together. Anyone is free to inspect and modify the source code (a bit like the recipe) that makes it work. You don't have to pay to use Linux, and you're allowed to share it with other people, too.

You probably won't be able to run the software you have on your other computers on your Raspberry Pi. It won't run Windows or Mac software, and not all Linux software works on the Raspberry Pi. But a lot of Linux software that is compatible with the Raspberry Pi is available and is free of charge.

## Figuring Out What You Can Do with a Raspberry Pi

The Raspberry Pi is a fully featured computer, and you can do almost anything with it that you can do with a desktop computer.

It has a graphical windows desktop to start and manage programs (see Chapter 4) as well as a shell for accepting text commands. (See Chapter 5.) You can use it for browsing the Internet (see Chapter 4), or for word processing and spreadsheets (see Chapter 6), or for editing photos. (See Chapter 7.) You can use it for playing back music or video (see Chapter 8) or for playing games. (See Chapter 19.) You can use the built-in software to write your own music, too. (See Chapter 14.) It's the perfect tool for homework, but it's also a useful computer for writing letters, managing your accounts, and paying bills online.

The Raspberry Pi is at its best, however, when it's being used to learn how computers work, and how you can create your own programs or electronics projects using them. It comes with Scratch (see Chapter 9), which enables people of all ages to create their own animations and games while learning some of the core concepts of computer programming along the way.

It also comes with Python (see Chapter 11), a professional programming language used by YouTube, Google, and Industrial Light & Magic (the special effects gurus for the *Star Wars* films), among many others.

It has GPIO pins on it that you can use to connect up your own circuits to the Raspberry Pi, so you can use your Raspberry Pi to control other devices and to receive and interpret signals from them. In Part 5, we show you how to build some electronic projects controlled by the Raspberry Pi.

## Getting Your Hands on a Raspberry Pi

One of the great things about the Raspberry Pi is that it's established a community of businesses that have created products for it, or have shared in its success by selling it. You can now buy the Raspberry Pi from a wide range of electronics companies for hobbyists. Global retailers include Pimoroni ([www.pimoroni.com](http://www.pimoroni.com)), The Pi Hut (<https://thepihut.com>), and Adafruit ([www.adafruit.com](http://www.adafruit.com)). It's also available from the Raspberry Pi's distributors, RS Components ([www.rs-components.com](http://www.rs-components.com)) and Element14 ([www.element14.com](http://www.element14.com)).

You might also be able to buy it from your local computer or electronics store, although you'll probably find it's only available as part of a kit there. Shops often bundle the Raspberry Pi with other items you need to use it. It can be convenient to get everything at once, but it might not represent the cheapest way to get started.

# Determining What Else You Need

The creators of Raspberry Pi have stripped costs to the bone to enable you to own a fully featured computer for less than \$35, so you'll need to scavenge or buy a few other bits and pieces in order to use your Pi. We say *scavenge* because the things you need are exactly the kind of things many people have lying around their house or garage already, or can easily pick up from friends or neighbors. In particular, if you're using a Raspberry Pi as your second computer, you probably have most of the peripherals you need. That said, you might find they're not fully compatible with the Raspberry Pi and you need to buy replacements to use with the Pi.

Here's a checklist of what else you might need:

- » **Monitor:** The Raspberry Pi has a high-definition video feed and uses an HDMI (high-definition multimedia interface) connection for it. If your monitor has an HDMI socket, you can connect the Raspberry Pi directly to it. If your monitor does not support HDMI, it probably has a DVI socket, and you can get a simple and cheap converter that enables you to connect an HDMI cable to it. Older VGA (video graphics array) monitors require a device to convert the HDMI signal into a VGA one. If you're thinking of buying a converter, check online first to see whether it works with the Raspberry Pi. A lot of cheap cables are just cables, when what you need is a device that converts the signal from HDMI format to VGA, not one that just fits into the sockets on the screen and your Raspberry Pi. These converters can be quite expensive, so Gert van Loo has designed a device that uses the Raspberry Pi's GPIO pins to connect to a VGA monitor. He's published the design specs so that anyone can build one, and sell it if she wants to, too. Take a look at eBay if you need one, and you might well find what you need. For more information, check out [www.raspberrypi.org/blog/gert-vga-adapter](http://www.raspberrypi.org/blog/gert-vga-adapter). (If your monitor is connected using a blue plug and the connector has three rows of five pins in it, it's probably a VGA monitor.)
- » **TV:** You can connect your Raspberry Pi to a high-definition TV using the HDMI socket and should experience a crisp picture. If you have an old television in the garage, you can also press it into service for your Raspberry Pi. The Pi can send a composite video signal, so it can use a TV as its display. When we tried this, it worked but the text lacked definition, which made it difficult to read. If a TV is your only option, see Appendix A for advice on tweaking the settings to get the clearest possible picture. It's better to use a computer monitor if you can, though. You'll need to get a cable with the right connector to fit your Pi: The Model A and Model B have a dedicated RCA video socket, but later models use the headphone socket for RCA video output, too.
- » **USB hub:** The Raspberry Pi has one, two, or four USB sockets (depending on the model you get). Consider using a powered USB hub, for two reasons.



Firstly (and especially if you have a Model A, A+, B, or Zero), you're going to want to connect other devices to your Pi at the same time as your keyboard and mouse, which need two sockets. And secondly, a USB hub provides external power to your devices and minimizes the likelihood of experiencing problems using your Raspberry Pi, especially if connecting relatively power-intensive devices such as hard drives. Make sure your USB hub has its own power source, independent of the Raspberry Pi.

- » **USB keyboard and mouse:** The Raspberry Pi only supports USB keyboards and mice, so if you're still using ones with PS/2 connectors (round rather than flat), you need to replace them.

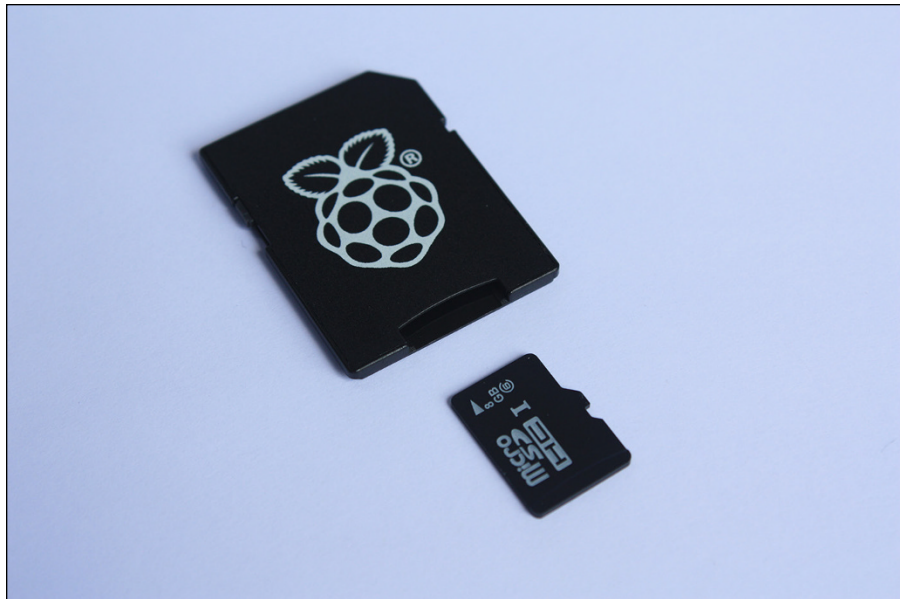


TIP

When the Raspberry Pi behaves unpredictably, it can be because the keyboard is drawing too much power, so avoid keyboards with too many flashing lights and features.

- » **MicroSD card or SD card:** The Raspberry Pi doesn't have a hard drive built into it, so it uses a MicroSD card (current models) or SD card (older models, excluding the Model B+) as its main storage. You probably have some SD cards that you use for your digital camera, although you might need to get a higher-capacity one. We recommend an 8GB card as a minimum, but you can use a 4GB card if you use a media center operating system (OS) like LibreELEC. (See Chapter 8 for a guide to LibreELEC.) Even that isn't much space compared to the hard drive on a modern computer, but you can use other storage devices such as external hard drives with your Raspberry Pi, too. SD and MicroSD cards have different class numbers that indicate how fast you can copy information to and from them. The Raspberry Pi Foundation recommends its branded Class 6 card as a good value and high-performing solution, although online retailers often sell a Class 10 card for this purpose. You will be fine with a Class 6 or higher. The easiest way to get started with the Raspberry Pi is to buy a card with the NOOBS software already on it. (See Chapter 2 for more on NOOBS.) Online retailers, including those mentioned earlier for buying the Pi itself, usually sell an 8GB or 16GB MicroSD card that has the NOOBS software preloaded on it. It comes with an SD card adapter, shown in Figure 1-3, so the card fits the older models (with the adapter) and newer models (without it).
- » **SD or MicroSD card writer:** Many PCs today have a slot for SD or MicroSD cards, so you can easily copy photos from your camera to your computer. If yours doesn't, you might want to consider getting an SD or MicroSD card writer to connect to your computer. You can use it to copy software to an SD card for use with your Raspberry Pi, but you won't be able to use it to copy files from your Raspberry Pi to a Windows computer. Alternatively, as we've said, you can buy a MicroSD card with an SD adapter that has the Raspberry Pi software already on it. You can also use the card writer to create a backup copy of your Raspberry Pi's files and software. (You can read about making back-ups in Chapter 4.)

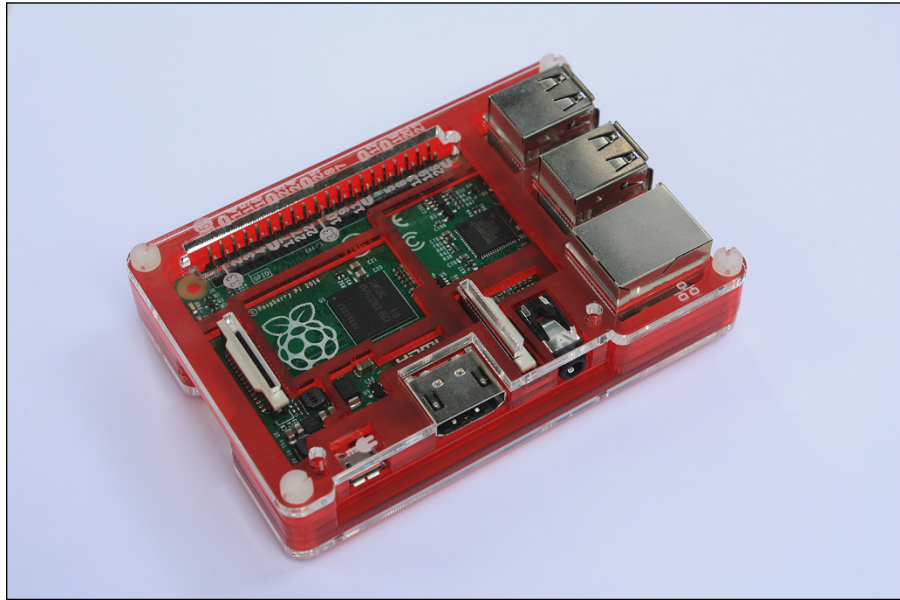
**FIGURE 1-3:**  
A NOOBS card. The MicroSD card fits into the SD card adapter. The MicroSD card works on the current Pi models, and inside the adapter it also fits the older Model A and Model B.



- » **USB key:** A USB key (also known as *flash drive* or *memory stick*) can be fairly cheap and high-capacity now (a 64GB USB key is readily affordable), which makes it an ideal complement to your Raspberry Pi. You can transfer files between your PC and your Raspberry Pi using a USB key, too.
- » **USB Wi-Fi adapter:** The Model A and A+ don't have an Ethernet socket, so if you want to connect to the Internet, you'll need a USB Wi-Fi adapter. You might already have one of these from a laptop. Some are incompatible with the Raspberry Pi, but companies that sell the Pi usually sell a compatible Wi-Fi adapter, too. There is an official Raspberry Pi Wi-Fi adapter, available from the usual Pi retailers.
- » **External hard drive:** If you want lots of storage, perhaps so that you can use your music or video collection with the Raspberry Pi, you can connect an external hard drive to it over USB. You'll need to connect your hard drive through a powered USB hub, or use a hard drive that has its own external power source.
- » **Raspberry Pi Camera Module:** The Raspberry Pi has stimulated entrepreneurs to create all kinds of add-ons for it, but the Camera Module is a product that originated at the Raspberry Pi Foundation. This 8-megapixel, fixed-focus camera can be used to shoot HD video and take still photos. There is also a version without an infrared filter (the PiNoIR Camera), which can be used for wildlife photography at night or weird special effects by day.

- » **Speakers:** The Raspberry Pi has a standard audio out socket, compatible with headphones and PC speakers that use a 3.5mm audio jack. You can plug headphones directly into it, or use the audio jack to connect to speakers, a stereo, or a TV. If you're using a TV or stereo for sound, you can get a cable that connects the 3.5mm audio jack and the audio input(s) on your television or stereo. You won't always need speakers: If you're using an HDMI connection, the audio is sent to the screen with the video signal, so you won't need separate speakers — but note that this doesn't work if you use a DVI monitor.
- » **Power supply:** The Raspberry Pi uses a Micro USB connector for its power supply, and is theoretically compatible with a lot of mobile phone and tablet chargers. In practice, many of these can't deliver enough current (up to 700 milliamperes for a Model A+ and up to 2500 milliamperes for a Raspberry Pi 3 Model B), which can make the Raspberry Pi perform unreliably. It's worth checking to see whether you have a 5V charger that might do the job (it should say on it how much current it provides), but for best results, we recommend buying a compatible charger from the same company that you buy your Raspberry Pi from. There is an official Raspberry Pi power supply available, which works in the U.K., Europe, the U.S./Japan, and Australia/China. For mobile applications, it's possible to power the Raspberry Pi using a battery pack designed for mobile phone charging. Don't try to power the Pi by connecting its Micro USB port to the USB port on your PC with a cable, because your computer probably can't provide enough power for your Pi. You can also power the Pi through the GPIO pins, but you could damage the Raspberry Pi if there is a spike in current or the wrong voltage is applied. If you want to provide power through the GPIO pins, a safer approach is to use a HAT device (*Hardware Attached on Top*) designed to sit on the GPIO pins and provide the consistent power you need while protecting the Pi underneath. The Raspberry Pi Foundation advises that you should only use batteries to power your Raspberry Pi if you know what you're doing, because there's a risk of damaging your Raspberry Pi. For more details on the power requirements of various Raspberry Pi models, consult the FAQ at [www.raspberrypi.org/help/faqs/](http://www.raspberrypi.org/help/faqs/).
- » **Case:** It's safe to operate your Raspberry Pi as is, but many people prefer to protect it from spills and precariously stacked desk clutter by getting a case for it. You can buy plastic cases on eBay ([www.ebay.com](http://www.ebay.com)), many of which are transparent, so you can still admire the circuitry and see the Pi's LED lights. These cases typically come supplied as simple kits for you to assemble. The PiBow Coupe (<https://shop.pimoroni.com/collections/pibow>) is one of the most attractively designed cases, assembled from layers of colored plastic. (See Figure 1-4.) It's designed by Paul Beech, who designed the Raspberry Pi logo. There are also official red-and-white cases for the Raspberry Pi 3 (compatible with the B+ and later models) and the Raspberry Pi Zero. The case for the Pi Zero includes three different tops, so you can either seal it, leave a camera hole, or have access to the GPIO pins. You don't have to buy a case,

though. You can go without or make your own using cardboard or Lego bricks. Whatever case you go with, make sure you can still access the GPIO pins so that you can experiment with connecting your Pi to electronic circuits and try the projects in Part 5 of this book.



**FIGURE 1-4:**  
The Pibow Coupe  
case for the  
Raspberry Pi 3,  
Pi 2, and  
Model B+.

» **Cables:** You'll need cables to connect it all up, too. In particular, you need an HDMI cable (if you're using an HDMI or DVI monitor), an HDMI-to-DVI adapter (if you're using a DVI monitor), an RCA cable (if you're connecting to an older television), an audio cable (if you're connecting the audio jack to your stereo), and an Ethernet cable (for networking on models with an Ethernet port). Note that current Raspberry Pi models send the RCA video signal through a 3.5mm jack (headphone socket), and earlier ones had a dedicated RCA socket. You will need a different cable, depending on which version of the Pi's design you have. If you have a Raspberry Pi Zero, you will also need a converter for the Mini HDMI socket, and for the Micro USB socket. You can get these cables from an electrical-components retailer and might be able to buy them at the same time as you buy your Raspberry Pi. Any other cables you need (for example, to connect to PC speakers or a USB hub) should come with those devices.



WARNING

The Raspberry Pi has been designed to be used with whatever accessories you have lying around, to minimize the cost of getting started, but in practice not all devices are compatible. In particular, incompatible USB hubs, keyboards, and mice can cause problems that are hard to diagnose. USB hubs that feed power back into your Raspberry Pi through the Pi's USB port (known as *backpowering*) could potentially cause damage to the Raspberry Pi if they feed in too much power.

A list of compatible and incompatible devices is maintained at [http://elinux.org/RPi\\_VerifiedPeripherals](http://elinux.org/RPi_VerifiedPeripherals), and you can check online reviews to see whether others have experienced difficulties using a particular device with the Raspberry Pi.

If you're buying new devices, you can minimize the risk by buying recommended devices from Raspberry Pi retailers.

In any case, you should set a little bit of money aside to spend on accessories. The Raspberry Pi is a cheap device, but buying a keyboard, mouse, USB hub, and cables can easily double or triple your costs, and you might have to resort to that if what you have on hand turns out not to be compatible.



- » Starting Scratch
- » Understanding the Scratch screen layout
- » Positioning and resizing your sprite
- » Making your sprite move
- » Changing your sprite's appearance
- » Adding sounds and music

## Chapter 9

# Introducing Programming with Scratch

**T**he Raspberry Pi was created partly to inspire the next generation of programmers, and Scratch is the perfect place to start. With it, you can make your own cartoons and games and discover some of the concepts that professional programmers use every day.

Scratch is designed to be approachable for people of all ages. The visual interface makes it easy to see what you can do at any time, without having to remember any strange codes, and you can rapidly achieve great results. Scratch comes with a library of images and sounds, so it takes only a few minutes to write your first Scratch program.

In this chapter, we introduce you to Scratch so that you can start to experiment with it. In Chapter 10, we show you how to use Scratch to make a simple arcade game.

# Understanding What Programming Is



Before we dip into Scratch, we should clear up some of the jargon surrounding it. A *program* is a repeatable set of instructions to make a computer do something, such as play a game. Those instructions can be extremely complicated because they have to describe what the computer should do in detail. Even a simple bouncing-ball game requires instructions for drawing the ball, moving it in different directions, detecting when it hits something, and then changing its direction to make it bounce.

*Programming* is the art and science of creating programs. You can create programs in lots of different ways, and Scratch is just one of them. In Chapter 11, you read about Python, another one.

Scratch and Python are both *programming languages*, different ways of writing instructions for the computer. Different programming languages are best suited for different tasks. Scratch is ideal for making games, for example, but it's not much use if you want to create a word processor. Using Python to create games takes longer, but it is more powerful than Scratch and gives you much more flexibility in the type of things you can get the computer to do.

## Working with Scratch

There are two versions of Scratch installed in Raspbian:

- » **Scratch:** This is the original version of Scratch, widely known as Scratch 1.4.
- » **Scratch 2:** This is an implementation of the newer version of Scratch, widely known as Scratch 2.0. If you use the online version of Scratch (at <https://scratch.mit.edu>), Scratch 2 is the version you are familiar with. Scratch 2 adds some features that are not available in the original Scratch, including the ability for sprites (which are, roughly speaking, game characters) to create copies of themselves, and a feature to make your own blocks (or instructions). (Both are described at the end of this chapter.) Because Scratch 2 requires the Flash software to work, it only runs on the Raspberry Pi 2 and Raspberry Pi 3. Some programs, especially action games, run significantly slower in this version of Scratch, because of all the layers of software required to run Flash on the Raspberry Pi. Simple programs will run fine, but if you want to make games, you will soon notice a performance difference.



Both versions of Scratch have been enhanced with features for controlling electronics projects with your Raspberry Pi, as you will see in Chapter 16.

If you're a newcomer to Scratch, we recommend you start by using the original Scratch, because your programs will run much more quickly. If you're already familiar with Scratch from the online version, or you're building programs that could use the additional features of Scratch 2, you might prefer to choose that version instead.

Because of the speed issues associated with Scratch 2, and the fact that it doesn't run on the Model B+ or earlier Pi models, we're going to use the original Scratch in our examples. We will, however, introduce you to both versions in this chapter so you can find your way around Scratch 2 as you build your own projects.



REMEMBER

You can take files you create in Scratch and use them in Scratch 2, but you can't take Scratch 2 files and open them in Scratch.

You access Scratch from the desktop environment.

To start Scratch, select your chosen version from the Applications menu in the top left of the screen. You can find Scratch and Scratch 2 in the `Programming` folder.

## Understanding the Scratch screen layout

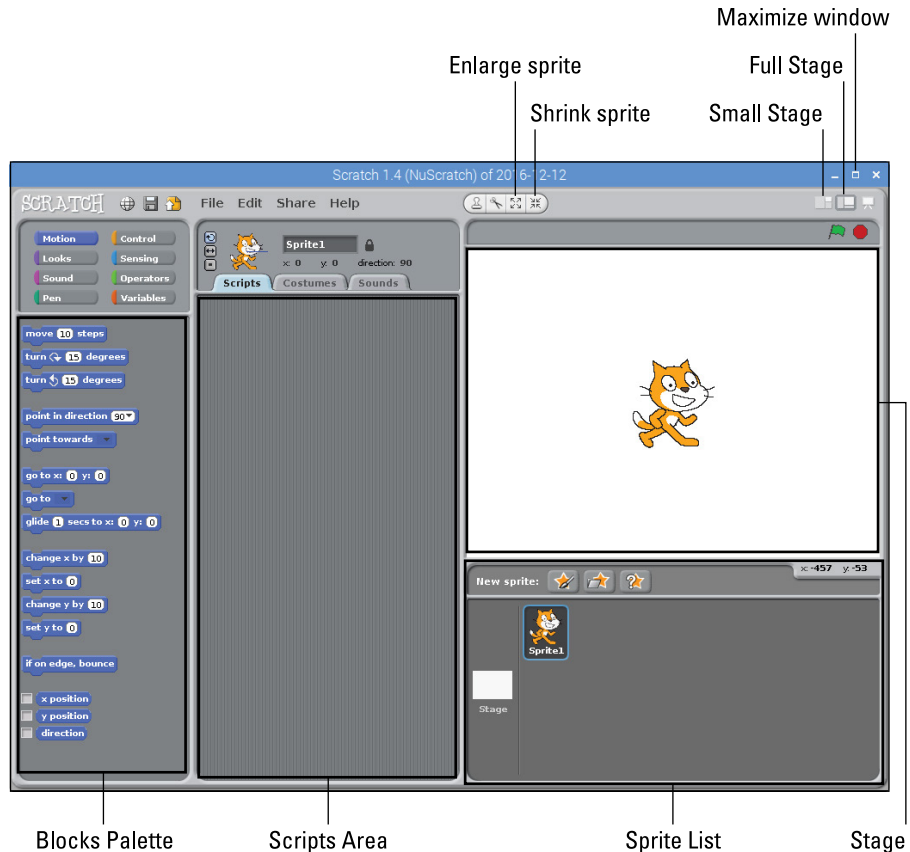
Scratch divides the screen into four main areas, as you can see in Figures 9-1 and 9-2. The Stage is where you can see your game or animation take shape. There's a cat on it already, so you can get started straightaway by making it do things, as you see in a minute. In Scratch, the Stage is in the top right, and in Scratch 2, it's in the top left.

Underneath the Stage is your Sprite List. You can think of sprites as the characters in your game. They're images that you can make do things, such as move around or change their appearance. For now, there's just the cat, which has the name `Sprite1` in Scratch and `Cat1` in Scratch 2.

You create a Scratch program by snapping together *blocks*, which are short instructions. You find the blocks in the Blocks Palette, which is on the left in Scratch and in the middle in Scratch 2. In both cases, it displays the Motion blocks by default. They include instructions to move ten steps, rotate, go to a particular grid reference, and point in a particular direction.

The Scripts Area is where the magic happens! You assemble your program in this space by dragging blocks into it from the Blocks Palette. The Scripts Area is in the

middle in the original Scratch and on the right in Scratch 2. In the original Scratch, you can use two buttons in the top right (refer to Figure 9-1) to toggle the Stage between full size and small. In Scratch 2, click the Edit menu to find the option for Small Stage Layout. When the Stage is small, the Scripts Area is bigger, so you might find that useful when you're writing scripts later in this chapter.



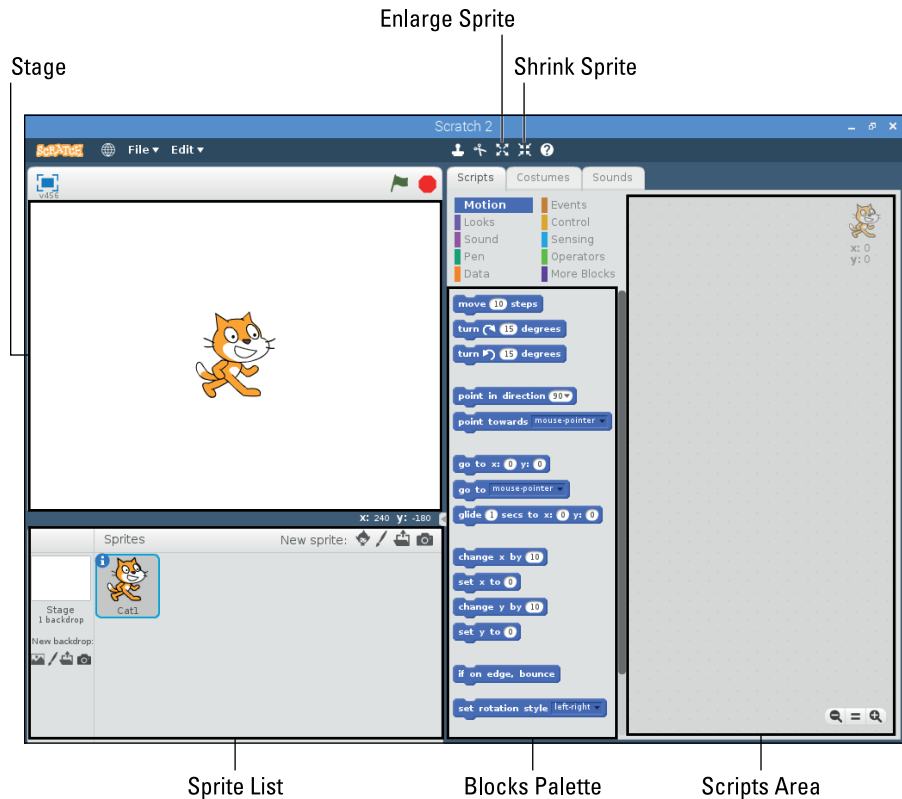
**FIGURE 9-1:**  
The screen layout  
in the original  
Scratch.

*Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See <http://scratch.mit.edu>.*

## Positioning and resizing your sprite

You can drag and drop your sprite (the cat) around the Stage to position it where you would like it to be at the start of your program.

You can also resize the sprite. Two buttons above the Stage (refer to Figures 9-1 and 9-2) are used to enlarge or shrink a sprite. Click one of them, and the mouse pointer changes to arrows pointing outward (for enlarging) or inward (for shrinking). Click your sprite on the Stage repeatedly to change its size.



**FIGURE 9-2:**  
The screen layout  
in Scratch 2.

*Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See <http://scratch.mit.edu>.*

When you've finished resizing, click something that isn't a sprite to return the mouse pointer to normal and stop resizing.

## Making your sprite move

Experimenting with Scratch is easy. To try out different blocks, just click them in the Blocks Palette. For example, try clicking the Move 10 Steps block, and you should see the cat move to the right. You can also turn the sprite 15 degrees in either direction by clicking the appropriate blocks.



**TIP**

If your cat goes somewhere that you don't want it to (don't they always?), you can click it on the Stage and drag it back to where you want it. You can reset rotation in the original Scratch, too, by clicking the tiny cat at the top of the Scripts Area, holding down the mouse button, and rolling the mouse in a circle pattern on the desk until the sprite is at the angle you want. In Scratch 2, you can correct a

sprite's rotation by clicking the sprite's *i* button in the Sprite List, and then clicking and dragging the blue line on the direction indicator. When you've finished, click the back arrow to show the Sprite List again.

Not all of the blocks will work at the moment because some of them need to be combined with other blocks. There's no harm in experimenting, however. If you click something that doesn't work, you might get an error message, but you won't cause any harm to Scratch or your Raspberry Pi.

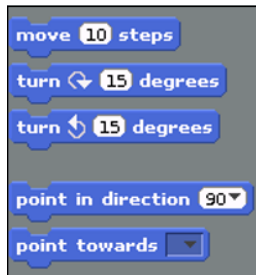
Next, we talk you through the different Motion blocks you can use.

## Using directions to move your sprite

You can use two different methods to position and move sprites. The first is to make your sprite “walk,” and to change its direction when you want it to walk the other way. The other is using grid coordinates, which we cover in the next section.

Here are the five blocks you use to move your sprite in this way (see Figure 9-3):

» **Move 10 Steps:** This makes your sprite walk in the direction it is facing. If your sprite has been rotated, the steps taken could move it in a diagonal line across the Stage. You can click the number in this block and then type another number to increase or decrease the number of steps taken, but bigger numbers spoil the illusion of animation. It stops looking like the sprite is walking across the screen when the number of steps taken is too big.

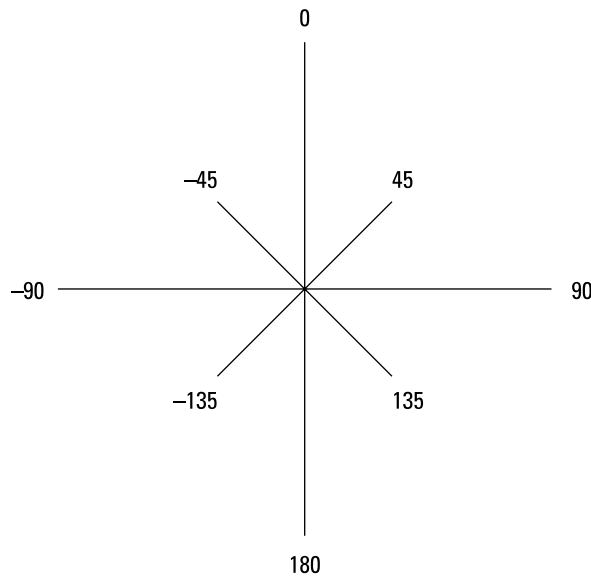


**FIGURE 9-3:**  
The directional-  
movement  
blocks.

*Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See <http://scratch.mit.edu>.*

» **Turn Right or Left 15 Degrees:** This block rotates your sprite. As with the number of steps, you can edit the number to change the degree by which your sprite is rotated. It walks in the direction it is facing when you use the Move 10 Steps block.

» **Point in Direction 90:** Whatever direction your sprite is facing, this block points it in the direction you want it to face. Use this block as is to reset your sprite to face right. You can change the number in this block to change the direction you want your sprite to face, and the numbers are measured in degrees from the position of facing up. (See Figure 9-4.) It helps to think of it like the hands of a clock: When the hand is pointing right, it's 90 degrees from the 12 o'clock position; when it's pointing down, it's 180 degrees from the top. To point left, you use -90. When you click the arrow in the right of the number box, it gives you a menu from which you can select the four main directions — but you can enter any number.



**FIGURE 9-4:**  
The number of degrees used to face in different directions.



**TIP**

You might be wondering whether you can enter **270** to point left, and the answer is that it works, but it can cause errors in your programs. If you turn your cat to direction 270 and then ask Scratch which way the cat is facing, it tells you -90. To avoid any inconsistencies like this, keep direction numbers in the range from -179 to 180.

» **Point Towards:** You can also tell the sprite to point toward the mouse pointer or another sprite. Use the menu in this block to choose what you would like your sprite to point toward.



REMEMBER

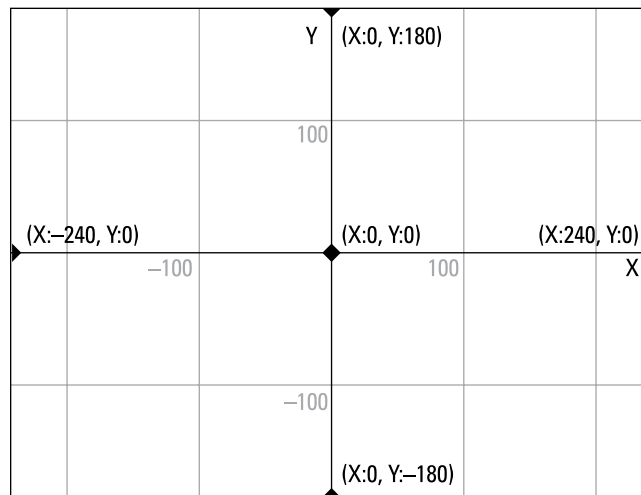
If you're changing the number value in a block, you still need to click the block to run it.

## Using grid coordinates to move and position your sprite

The second way you can move and position your sprite is to use grid coordinates. That makes it easy to position your sprite at an exact place on the screen, irrespective of where it is now.

Every point on the Stage has two coordinates: an X position (indicating where it is horizontally) and a Y position (indicating where it is vertically). The X positions are numbered from  $-240$  at the far left to  $240$  at the far right. The Y positions are numbered from  $-180$  at the bottom edge of the Stage to  $180$  at the top edge. That means the Stage is a total of  $480$  units wide and  $360$  units tall. The center point of the screen, where your cat begins its day, is where X equals  $0$  and Y equals  $0$ . Figure 9-5 provides a quick visual reference of how the coordinates work.

When you move the mouse over the Stage, the grid reference of your mouse pointer is shown just underneath the Stage on the right.

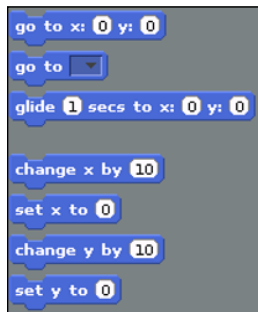


**FIGURE 9-5:**  
The grid coordinates on the Stage.

Seven Motion blocks use the X and Y coordinates (see Figure 9-6):

- » **Go to x:0 y:0:** You can use this block to position your sprite at a specific point on the Stage. By default, it returns a sprite to the center of the screen ( $x=0$ ,  $y=0$ ). Edit the numbers for X and Y to position your sprite somewhere else.

- » **Go to:** Use this block to move your sprite to the mouse pointer's location, or to the location of another sprite, if you have more than one.
- » **Glide 1 secs to x:0 y:0:** When you use the Go To block, your sprite just jumps to its new position. The Glide block makes your sprite float there smoothly instead. You can change the number of seconds the glide takes, including using decimals for part of a second (for example, 0.5 for half a second).
- » **Change X by 10:** This moves your sprite ten units to the right. You can change the number of units and use a negative number if you want to move left instead. Note that this doesn't affect your sprite's vertical position and is independent of which way around your sprite is facing.
- » **Set X to 0:** This changes the horizontal position of your sprite on the Stage, without affecting its vertical position. The value 0 returns it to the center of the screen horizontally, and you can edit the number to position it left or right of that. Use a negative number for the left half of the screen and a positive number for the right half.
- » **Change Y by 10:** This moves your sprite ten units up the Stage, without affecting its horizontal position, and irrespective of which direction it is facing. You can change the number of units and use a negative number to move the sprite down the screen instead.
- » **Set Y to 0:** This changes the vertical position of your sprite on the Stage without affecting its horizontal position, and without regard to which way it faces. Use a positive value for the top half of the Stage and a negative value for the lower half.



**FIGURE 9-6:**  
The blocks used  
for moving  
sprites using grid  
coordinates.

*Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See <http://scratch.mit.edu>.*



REMEMBER

You need to run a block to actually see its effect on your sprite. Do this by clicking it.

## Showing sprite information on the Stage

It can be hard to keep track of where your sprite is and in which direction it's facing, but you can show the values for its X position, Y position, and direction on the Stage. Select the boxes at the bottom of the Blocks Palette to do this, as shown in Figure 9-7. They clutter the screen a bit, but they can be essential tools for testing when you're creating a game.



**FIGURE 9-7:**  
The blocks used  
to show sprite  
information on  
the Stage.

*Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See <http://scratch.mit.edu>.*

## Changing your sprite's appearance

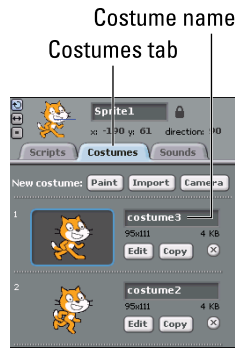
As well as moving your sprite around the screen, you can change what it looks like.

### Using costumes

One way to think of sprites is as the characters in a game (although they can be used for lots of other objects, too, such as obstacles). Each sprite can have a number of *costumes*, which are different pictures of it. If the costumes look fairly similar, you can create the illusion of animation by switching between them. Your cat sprite comes with two costumes, and when you switch between them, it looks like the cat is running. You can think of a costume as being one image in an animation sequence (an *animation frame*).

You can see the costumes for your sprite by clicking the Costumes tab at the top of the Scripts Area, as shown in Figure 9-8. If you want to modify the cat's appearance, you can click the button to edit one of the costumes in Scratch or simply click the sprite in Scratch 2 and use the editing canvas to its right. If you want to create a new animation frame, you can click the Copy button beside a costume in Scratch, or in Scratch 2 you can right-click the costume and choose duplicate from the menu that opens. You can then edit the bits you want to change.





**FIGURE 9-8:** You can change a sprite's appearance by giving it a new costume. (Image shows the original Scratch.)

*Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See <http://scratch.mit.edu>.*



TIP

It doesn't matter much when you're experimenting with sprites, but when you make your own games and animations, you can save yourself a lot of brain ache by giving your sprites meaningful names. It's much easier to remember that the costume with the name *game over* should be shown when the player is defeated than it is to remember it's called *costume7*. To rename a costume, click the Costumes tab to show the costumes, and then click the costume's current name (refer to Figure 9-8) and type its new name. In Scratch 2, the costume's name is shown above the editing canvas.

In the Blocks Palette, there are two blocks you can use to switch between costumes. (See Figure 9-9.) Click the Looks button above the Blocks Palette to show them:

- » **Switch to Costume:** If you want to switch to a particular costume, choose its name from the menu in this block and then click the block.
- » **Next Costume:** Each time you use this block, the sprite changes to its next costume. When the costumes run out, it goes back to the first one again.



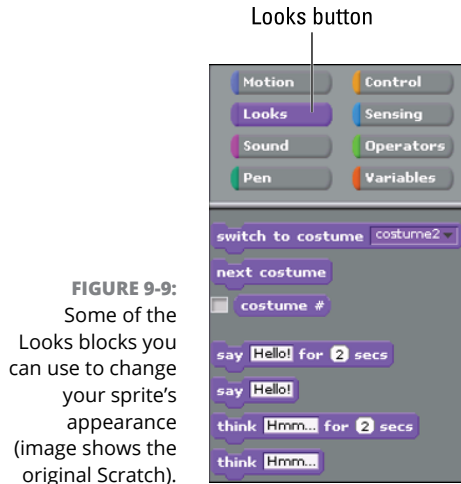
TIP

You can show a sprite's costume number on the Stage, too, so that it's easier for you to work out what's going on. Just select the check box next to Costume # in the Blocks Palette.

## Using speech and thought bubbles

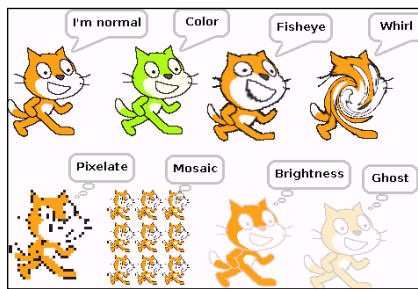
Scratch includes four blocks you can use to show a speech bubble or a thought bubble onscreen. (Refer to Figure 9-9.) To see them, and to see the other blocks that change a sprite's appearance, click the Looks button above the Blocks Palette. The speech and thought bubbles are great for giving a message to the player or

viewer. You can edit the word in the block (*Hello!* or *Hmm...*) to change the text in the bubble. Figure 9–10 shows the speech bubbles (top row) and thought bubbles (bottom row) in action.



**FIGURE 9-9:** Some of the Looks blocks you can use to change your sprite's appearance (image shows the original Scratch).

*Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See <http://scratch.mit.edu>.*



**FIGURE 9-10:** The different graphic effects you can apply to your sprite.

*Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See <http://scratch.mit.edu>.*

If you use one of the options with a length of time in it, the sprite pauses for that length of time and the bubble disappears when it has elapsed.

If you use a block without a length of time, you can make the bubble disappear again by using the Say or Think block again but editing the text so that the text box in the block is empty.

## Using graphic effects

You can apply several graphic effects to your sprite using Looks blocks. In Figure 9-10, we've used eight sprites to demonstrate them on the Stage. The Color effect changes the sprite's Color Palette, turning orange to green in the case of the cat. The Fisheye effect works like a fish-eye lens, making the central parts of the sprite appear bigger. Whirl distorts the sprite by twisting its features around its middle. Pixelate makes the sprite blocky. Mosaic shrinks the sprite and repeats it within the space it usually occupies. The Brightness and Ghost effects can sometimes look similar, but the Brightness effect increases the intensity of the colors (turning the cat's black outline silver while softening the orange), and the Ghost effect fades out all colors evenly.

Here are the three blocks you use to control graphic effects:

- » **Change Color Effect by 25:** You can select the effect you want to change (by default, it's the color effect) and enter the amount of it you want to add. You can use negative numbers to reduce the extent to which the effect is applied to your sprite. The color effect has 200 different levels (from 0 to 200), and the other effects typically look best with levels in the range from -100 to 100. Experiment!
- » **Set Color Effect to 0:** Use this block to set a chosen effect to a specific level. Choosing 0 turns the effect off again. You can use any of the seven effects with this block.
- » **Clear Graphic Effects:** This block removes all graphic effects you've applied to a particular sprite so that it looks normal again.



The graphic effects look great, but they are quite slow on a Raspberry Pi Model B+ or older. They're best used in moderation for special moments in your animation or game; otherwise, they might make it appear unresponsive.

## Resizing your sprite

Earlier in this chapter, we show you how to change the starting size of your sprite on the Stage. You can use blocks to issue instructions to change its size, too, so you could make it grow larger as the game progresses, for example.

There are two blocks you can use to resize your sprite:

- » **Change Size by 10:** This block enables you to change the size of your sprite by a certain number of units, relative to its current size. As usual, you can edit the number. If you want to decrease the sprite's size, use a negative number.

» **Set Size to 100%:** This block sets the size to a percentage of its original size, so with the default value of 100 percent, it effectively resets any resizing you've done.



TIP

You can also select the check box beside the Size block to show the sprite's size on the Stage, in the same way you display other sprite information there. (See "Showing sprite information on the Stage," earlier in this chapter.) This can be useful for testing purposes.

## Changing your sprite's visibility

Sometimes, you might not want your sprite to be seen on the Stage. If a spaceship is blown up in your game, for example, you want it to disappear from view. These two blocks give you control over whether a sprite is visible:

» **Hide:** Use this block to make your sprite invisible on the Stage. If a sprite is hidden, Scratch won't detect when it touches other sprites, but you can still move a hidden sprite's position on the Stage so that it's in a different place when you show it again.

» **Show:** By default, your sprite is visible, but you can use this block to reveal it again after you have hidden it.



TIP

Sometimes, sprites might get on top of each other. You can use the Go to Front block to make a sprite appear on top of all the others. To push a sprite backward and allow others to appear on top of it, use the Go Back 1 Layers block.

## Adding sounds and music

As well as changing a sprite's appearance, you can give it some sound effects. Scratch comes with sounds, including slurps, sneezes, and screams; ducks, geese, and owls; and pops, whoops, and zoops. You can find effects for most occasions, and many of them are natural partners for one of the sprites that Scratch provides.

To add a sound to your sprite, you have to do one task first: Import the sound to your sprite. Here's how you'd do that:

1. **Click the Sounds tab above the Scripts Area, as shown in Figure 9-11, and then click the Import button.**

In the original Scratch, it's labelled Import, but in Scratch 2, the button looks like a tiny speaker: Don't confuse it with the large speaker used to represent any existing sound effect on the sprite.



**FIGURE 9-11:** Adding sound effects to your sprite (Image shows original Scratch).

*Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See <http://scratch.mit.edu>.*

2. In the file browser that appears, browse the provided sounds.
3. (Optional) In Scratch, click a file once to hear a preview of it. In Scratch 2, click the play button beside the file to hear it.
4. Click a file twice to bring it into your sprite.
5. (Optional) After you've imported a sound, you can preview it. Click the speaker beside it in Scratch, or the play button in the Sounds Area in Scratch 2. In both versions of Scratch, click the X button to delete a sound from your project.

If you delete a sound in this way, it remains on your SD card so that you can import it again later.

After a sound has been imported, you use one of the Sound blocks to play a sound. To see all available Sound blocks, click the Sound button at the top of the Blocks Palette first.

The Play Sound block enables you to choose which sound you'd like to play from those you have imported. The Play Sound Until Done block stops the program from running any blocks joined underneath this one until the sound has finished playing.



REMEMBER

The sound is imported to a particular sprite, so if you can't see it as one of the choices in the Play Sound block, be sure you've imported it to the correct sprite. In Chapter 10, we cover how to use multiple sprites in a project.

There are also blocks that let you use virtual drums and pitched instruments to create music using Scratch. Notes are numbered: C is 60, C# is 61, D is 62, and so on. There's a block called Play Note 60 for 0.5 Beats that plays a note with a particular number for a certain duration. When you click the menu in this block to specify which note to play, a piano opens that you can use to select the note.



TIP

If you're new to music, you can generally get a good result by starting with C, sticking to the white notes, and making sure no two consecutive notes are too far apart on the piano.

You can use a block called Set Instrument to 1 to change the instrument, but if you want to use this, you need to install the different instruments first if you're using the original Scratch. To do so, click the Terminal icon on the task bar at the top of the desktop and then issue these commands in the Terminal window (explained in greater depth in Chapter 5):

```
sudo apt-get update
/usr/share/scratch/timidityinstall.sh
```

Finally, in order to finish installing the instruments, you need to reboot your Raspberry Pi. Save your work in any open applications first, and then open the Applications menu, click Shutdown, and choose Reboot.



TIP

The note numbers used in Scratch are the same as those used in Sonic Pi. (See Table 14-1, in Chapter 14.)

## Creating scripts

Clicking blocks in the Blocks Palette is one way to issue commands to Scratch, but if that's all you're doing, you're not really programming. The fact is, if you have to click each block every time you want to run it, you're doing all the hard work of remembering the instructions, and the computer can work only as fast as you can click the blocks.

A *program* is a reusable set of instructions that can be carried out (or *run*) whenever you want. To start to create a program, you drag blocks from the Blocks Palette and drop them in the Scripts Area in the middle of the screen. Most blocks mentioned so far in this chapter have a notch on the top and a lug on the bottom, so they fit together like jigsaw pieces. You don't have to align them perfectly: Scratch snaps them together for you if they're close enough when you release the mouse button.

You put your blocks in the order you want Scratch to run them, starting at the top and working your way down. It's a bit like making a to-do list for the computer.

A group of blocks in the Scripts Area is called a *script*, and you can run the script by clicking anywhere on it. Its border is highlighted, and you'll see the cat move around the Stage as you've instructed it to.

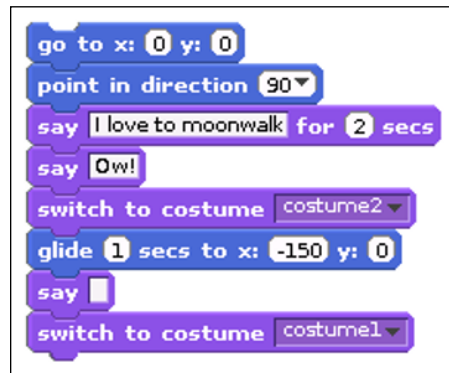
You can have multiple different scripts in the Scripts Area, so you could have one to make the cat walk left and another to make it walk right, for example. When you add multiple sprites (see Chapter 10), each sprite has its own Scripts Area and scripts there to control it.



TIP

If you want to tidy up the Scripts Area, you can move a script by dragging its top block. If you drag a block that is lower down in the script, it is separated from the blocks above it and carries with it all of the blocks below it. If you want to delete a block or set of blocks, drag it back to the Blocks Palette on the left.

The *moonwalk* is the dance, popularized by Michael Jackson, that makes you look like you're walking forward while you're actually moving backward. Figure 9-12 shows a script to make your cat moonwalk across the Stage. The first two lines in the script reset the cat to the middle of the screen, facing right. The cat tells you that it loves to moonwalk and then lets out a little whoop like Michael Jackson's, which it continues for the duration of the dance. The costume switch changes the position of the cat's legs, and it then glides 150 units to the left. Close the speech bubble by using the Say block with nothing in it, and then switch back to the other costume, which makes the cat's legs move back to their default position. Give it a go!



**FIGURE 9-12:**  
This is how you  
make a cat  
moonwalk. Ow!

*Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See <http://scratch.mit.edu>.*

## Using the Wait block to slow down your sprite

As you put your script together, you might find that some movements happen so fast that you can hardly see what's going on.

If you click the Control button at the top of the Blocks Palette, you can find a set of yellow blocks that are used to govern when particular things happen. You can read more about these in Chapter 10, but for now it's worth knowing that a block here enables you to wait for a certain number of seconds. Drag this into your script where necessary to introduce a delay so that you can see each of your blocks in action. The length of the delay is 1 second by default, but you can change it to whatever you want, including parts of a second (for example, 0.5 for half a second).



TIP

The Say Hello! for 2 Secs block can be also be used to force the script to pause before running any more blocks.

## Saving your work

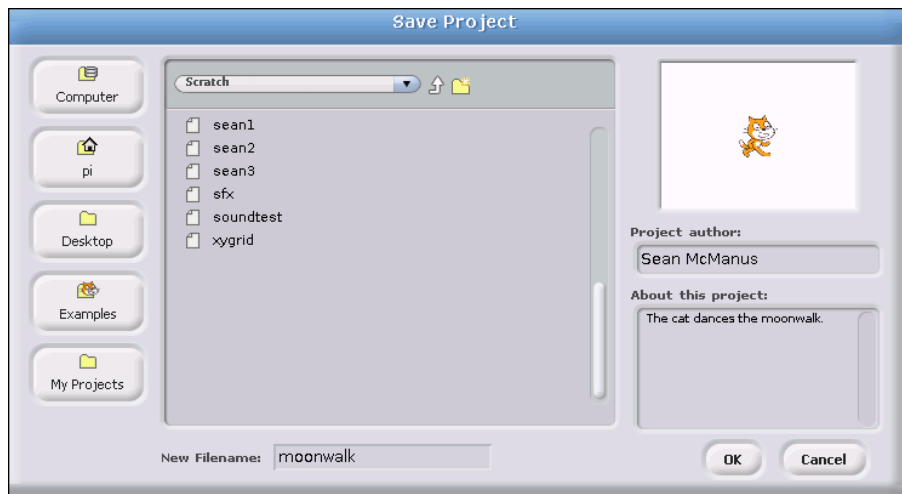
Remember to save your work so that you can come back to it later. You can find the option to save on the File menu, at the top of the screen, or you can click the Floppy Disk icon in the top left in the original Scratch.

In the original Scratch, when the Save Project dialog box opens (see Figure 9-13), you see buttons on the left to choose from various places you could save your file, although you might not have permission to use all of them. (See Chapter 5 for more on permissions.) We recommend that you use the *Scratch* folder inside your *pi* directory. On the right, you can add your name and some project notes to remind you later what the project was about. You can see and edit the project notes associated with a file by using the File menu when you're working on a program.

When you save in Scratch 2, you see a file browser that enables you to choose where to save. Again, we recommend saving in your *Scratch* folder inside your *pi* directory.



**FIGURE 9-13:** Saving your work so that you can come back to it later (image shows the original Scratch).



*Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See <http://scratch.mit.edu>.*

## What's new in Scratch 2

The two versions of Scratch on your Raspberry Pi share the same core blocks, but there are a few differences in Scratch 2 to be aware of:

- » **Events blocks:** This is a new category of blocks introduced in Scratch 2. It includes blocks that were categorized as Control blocks in the older version of Scratch. If the block name starts with “when” or “broadcast,” you’ll probably find it in the Events part of the Blocks Palette in Scratch 2. (The exception is When I Start As a Clone, which remains a Control block.)
- » **Cloning:** There’s a new feature in Scratch 2 to enable sprites to create copies (or clones) of themselves. Those clones can then run scripts that start when the clone is created. You can find the blocks to experiment with cloning in the Control part of the Blocks Palette.
- » **More Blocks:** Scratch 2 enables you to build your own blocks by combining existing blocks. It’s a great way to make programs that are easier to read. Click the More Blocks button above the Blocks Palette to try it.
- » **Data:** The Variables part of the Blocks Palette has been renamed to Data in Scratch 2. It is otherwise the same. We cover variables in Chapter 10.

Despite these differences, you should find that programs written for the original Scratch will work in Scratch 2, so you can use tutorials and books based on them. For more guidance on Scratch 2, see our recommended resources at the end of Chapter 10.



# Index

## Symbols and Numerics

- ' (apostrophe), 105
- \* (asterisk) operator, 105, 109, 211
- \ (backslash), 105
- : (colon), 268, 271
- { } (curly braces), 105, 230
- . (current directory), 110, 111
- \$ (dollar sign), 88
- && (double ampersand), 116
- == (double equal sign), 228
- // (double slash) operator, 211
- # (hash mark), 213, 450, 451
- (hyphen) operator, 88, 97, 100–101, 105, 211
- ... (parent directory), 90–91, 110, 111
- % (percent) operator, 211
- | (pipe) character, 121
- + (plus) operator, 211
- ? (question mark) wildcard, 105, 109
- " (quotation (speech) marks), 105
- << (shift operator), 366
- / (slash) operator, 105, 211
- [] (square brackets), 105, 245
- ~ (tilde) symbol, 90–91, 94–95
- 13.56 MHz tags, 390
- 125-135 KHz tags, 390

## A

- absolute paths, 94–96
- AC (alternating current), 302
- accepting user input, 216
- accessing
  - bookmarks, 77
  - Scratch, 165
- accessories, 16–21
- Acorn, 47
- active tags, 390
- actors, animating, 245–247

- Adafruit (website), 15, 325
- adapting
  - appearance of sprites in Scratch, 172–176
  - audiobook, 438
  - background in Scratch, 185
  - colors in GIMP, 145
  - desktop wallpaper, 60
  - directories in Linux Shell, 89
  - displays for files, 72–73
  - game speed, 204
  - parent directory in Linux Shell, 90–91
  - perspective in Minecraft, 258
  - screen display, 452–454
  - sensitivity of keyboard/mouse, 83
  - settings, 448–454
  - settings in media center, 157–158
  - sprite visibility in Scratch, 176
- Add Tab (File Manager), 68
- adding
  - blocks in Minecraft, 263–264
  - bookmarks, 67, 76–77
  - ceilings in Minecraft, 274
  - clouds, 248–250
  - media to media center, 151–155
  - music, 152–153
  - music in Scratch, 176–178
  - pictures, 154
  - scripts to Stage, 203
  - sounds in Scratch, 176–178
  - special effects in Sonic Pi, 292
  - sprites to games, 185–186
  - timers, 253
  - users, 119
  - videos, 153–154
- additive mixing, 359
- add-ons, 155
- addressable LEDs, 362–365

- adjusting
  - appearance of sprites in Scratch, 172–176
  - audiobook, 438
  - background in Scratch, 185
  - colors in GIMP, 145
  - desktop wallpaper, 60
  - directories in Linux Shell, 89
  - displays for files, 72–73
  - game difficulty, 253–254
  - game speed, 204
  - parent directory in Linux Shell, 90–91
  - perspective in Minecraft, 258
  - random number seed, 287–288
  - screen display, 452–454
  - sensitivity of keyboard/mouse, 83
  - settings, 448–454
  - settings in media center, 157–158
  - sprite visibility in Scratch, 176
- advanced listing options, 96–99
- advanced settings, 454
- Akerman, Dave (developer), 9, 441
- altering
  - appearance of sprites in Scratch, 172–176
  - audiobook, 438
  - background in Scratch, 185
  - colors in GIMP, 145
  - desktop wallpaper, 60
  - directories in Linux Shell, 89
  - displays for files, 72–73
  - game speed, 204
  - parent directory in Linux Shell, 90–91
  - perspective in Minecraft, 258
  - screen display, 452–454
  - sensitivity of keyboard/mouse, 83
  - settings, 448–454
  - settings in media center, 157–158
  - sprite visibility in Scratch, 176
- alternating current (AC), 302
- animate() function, 246
- animating actors, 245–247
- anode, 302
- APA102C protocol
  - about, 362–365
  - bit-banging, 365–366
- apostrophe ('), 105
- appearance, changing of sprites in Scratch, 172–176
- append() function, 224, 271
- applications, finding and installing, 83–84
- Applications folder, 27
- Applications menu, 60–63
- apropos command, 121
- apt, 116
- apt-cache, 115
- apt-get, 121
- arcade game, programming
  - about, 183–184, 205
  - adding scripts to Stage, 203
  - adding sprites, 185–186
  - adjusting game speed, 204
  - changing background, 185
  - controlling scripts, 190–195
  - deleting sprites, 184–185
  - detecting when sprites hit, 196–197
  - drawing sprites, 186–189
  - duplicating sprites, 203
  - fixing bugs, 201–203
  - making sprites move automatically, 199–200
  - naming sprites, 189
  - playing game, 204
  - starting new project, 184–185
  - using random numbers, 195–196
  - variables, 197–199
- arguments, 226
- ARM processor, 14
- asterisk (\*) operator, 105, 109, 211
- Audacity, 411
- audio. *See also* media center
  - about, 149
  - connecting, 44–45
  - fixing problems, 454–455
- audio socket, 45
- audiobook player, 437–438
- [a-z] wildcard, 109

**B**

- background, changing in Scratch, 185
- backing up data, 84–85

- backpowering, 21
- backslash (\), 105
- ball grid array (BGA) package, 308
- battery power, 46
- BCM2835, 308
- Beech, Paul (designer), 19
- Beneath a Steel Sky, 434–435
- BGA (ball grid array) package, 308
- bin directory, 92
- binary number, 347
- bit-banging
  - about, 365
  - APA102C protocol, 365–366
- Blinkt! board, 380
- blocks
  - adding in Minecraft, 263–264
  - breaking in Minecraft, 259–260
  - creating music with, 176–178
  - positioning in Minecraft, 259
  - in Scratch, 165–166
  - in Scratch 2, 181
- Blocks Palette (Scratch), 165–166, 173, 201
- Bluetooth devices, configuring, 54
- bookmarks, 67, 76–77
- boot directory, 92
- Brain party, 436
- breaking blocks in Minecraft, 249–260
- Brightness and Contrast settings (GIMP), 145
- Brightness effect (Scratch), 175
- Broadcast block, 193–194
- Broadcom BCM2835 chip, 14
- browsing, web
  - about, 74
  - bookmarks, 76–77
  - privacy, 77–78
  - searching within web pages, 76
  - tabbed browsing, 76
- Bryan, David (developer), 439–440
- bugs, fixing, 201–203
- building
  - chatbot program, 221–238
  - classes, 366–370
  - colors, 359–362

- computer-controlled dice, 337–347
- dictionary look-up function, 231–236
- directories in Linux Shell, 106
- files using redirection in Linux Shell, 104–105
- files/folders, 71
- folders, 71
- functions in Minecraft, 270–271
- graphics, 413–423
- invitations in LibreOffice Draw, 136–138
- main conversation loop, 236
- main loop in Minecraft, 271–274
- NOOBS card, 25–32
- presentations in LibreOffice Impress, 134–136
- programs, 241–255
- scripts in Scratch, 178–179
- sound samples, 411–413
- times tables program, 212–221
- variables, 197
- bytes, 342

## C

- cables
  - about, 20
  - replacing, 447
- cache, updating, 114
- cal command, 104
- calculating
  - circuit values, 306
  - sums with Python Shell, 210–212
- calling functions, 236
- canceling Linux commands, 102
- Canvas (Scratch), 188
- case, 19–20
- case-sensitivity
  - of Linux, 89
  - of Python, 10, 261
- Cat sprite, 184
- cathode, 302
- cd directory, 89, 90–91
- ceilings, adding in Minecraft, 274
- cellsVisitedList[] variable, 269
- changeover switch, 303

- changing
  - appearance of sprites in Scratch, 172–176
  - audiobook, 438
  - background in Scratch, 185
  - colors in GIMP, 145
  - desktop wallpaper, 60
  - directories in Linux Shell, 89
  - displays for files, 72–73
  - game difficulty, 253–254
  - game speed, 204
  - parent directory in Linux Shell, 90–91
  - perspective in Minecraft, 258
  - random number seed, 287–288
  - screen display, 452–454
  - sensitivity of keyboard/mouse, 83
  - settings, 448–454
  - settings in media center, 157–158
  - sprite visibility in Scratch, 176
- Chapellier, Cyril (developer), 443
- chatbot program
  - about, 221–222, 236–237
  - adding while loop, 227–228
  - creating dictionary look-up function, 231–236
  - creating main conversation loop, 236
  - dictionaries, 229–231
  - final listing, 237–238
  - forcing replies from players, 228–229
  - lists, 222–227
- Cheat Sheet (website), 4
- checking
  - connections, 446
  - file types in Linux Shell, 89–90
  - MicroSD card, 447
  - SD card, 447
  - what's installed, 118
- checkRequest function, 355
- chgrp command, 101
- chmod command, 101
- choose() method, 289
- choosing
  - files/folders, 70, 108–110
  - multiple files using wildcards, 108–110
- chord names, using, 285–286
- Chromium, web browsing with, 74–76
- circuits
  - about, 299–300
  - components, 307–308
  - connecting together, 326–327
  - creating computer-controlled dice, 337–347
  - electricity, 300–306
  - GPIO, 308–317
  - Pedestrian Crossing project, 347–356
  - ready-made add-on boards, 317–320
  - testing with simulators, 307
  - your first, 327–337
- classes, creating, 366–370
- Claws Mail, 61, 78
- Clear button (Scratch), 188
- Clemens, Michael (developer), 438
- clock signal, 365
- Clone tool (GIMP), 146–147
- cloning, in Scratch 2, 181
- closing program windows, 63–64
- Cloudbusting game
  - adding clouds, 248–250
  - adding timer, 253
  - adjusting difficulty, 253–254
  - animating actors, 245–247
  - collecting sounds and images, 240–241
  - creating programs, 241–255
  - detecting mouse clicks, 244–245
  - enabling multiple clouds to be clicked, 251–252
  - final listing, 254–255
  - regenerating clouds, 250–251
  - running programs, 241–255
  - setting up folders, 241
  - using random numbers, 247–248
- clouds
  - adding, 248–250
  - enabling multiples to be clicked, 251–252
  - regenerating, 250–251
- collidepoint() function, 245
- colon (:), 268, 271
- Color effect (Scratch), 175
- Color Palette (Scratch), 175
- colors
  - adjusting in GIMP, 145
  - creating, 359–362

- Colour Balance settings (GIMP), 145
- columns, resizing in LibreOffice Calc, 133
- commands
  - apropos, 121
  - cal, 104
  - chgrp, 101
  - chmod, 101
  - cp, 111–113
  - date, 104
  - echo, 104, 122
  - else, 236
  - entering, 103
  - file, 89–90, 110
  - groups, 118
  - input, 220
  - less, 102–103, 121
  - Less, 102–103
  - license(), 209
  - Linux, 102, 120–124
  - ls, 97–99
  - mc.player.getTilePos(), 279
  - mkdir, 106, 107, 121
  - passwd, 120
  - pidel, 123
  - ping, 457
  - print, 209, 210–212, 213–215, 216–218, 219
  - Python, 208–210
  - rm, 107
  - rmdir, 110
  - sleep, 284, 291–292
  - sync, 293
  - type, 120
  - useradd, 119
  - which, 120
  - while, 229
- commas, in Sonic Pi, 216
- common ground/reference, 310
- compatible devices, 21
- components, of circuits, 307–308
- composing random tunes using shuffle method, 287
- composite video socket, 42–43
- Compute Model, 14
- computer-controlled dice, creating, 337–347
- conditional statement, 235
- config.txt file, opening, 449
- configuring
  - Bluetooth devices, 54
  - Raspberry Pi in Raspbian, 50–53
  - Wi-Fi, 53
- connecting
  - about, 35–37
  - audio, 44–45
  - checking connections, 446
  - configuring Wi-Fi, 53
  - inserting SD card, 37–39
  - keyboard, 44
  - logging in, 50
  - monitor, 41–43
  - mouse, 44
  - power, 46–50
  - Raspberry Pi Camera Module, 39–41
  - router, 45–46
  - testing Raspberry Pi Camera Module, 54–55
  - TV, 41–43
  - USB hub, 43–44
  - using SSH, 458
- Control block, 190–191
- controlling
  - scripts, 190–195
  - software, 113–118
  - user accounts, 118–120
- conventional current, 302
- conversation loop, creating, 236
- converter chips, 381
- converting images between formats in GIMP, 147
- Cool Scratch projects in Easy Steps* (McManus), 205
- coordinates, Minecraft, 261–262
- copying
  - files in Linux Shell, 111–113
  - files/folders, 69
  - NOOBS to SD or MicroSD card, 30–32
- costumes, in Scratch, 172–173
- countdown() function, 253
- Cox, Simon (professor), 9
- cp command, 111–113

- creating
  - chatbot program, 221–238
  - classes, 366–370
  - colors, 359–362
  - computer-controlled dice, 337–347
  - dictionary look-up function, 231–236
  - directories in Linux Shell, 106
  - files using redirection in Linux Shell, 104–105
  - files/folders, 71
  - functions in Minecraft, 270–271
  - graphics, 413–423
  - invitations in LibreOffice Draw, 136–138
  - main conversation loop, 236
  - main loop in Minecraft, 271–274
  - NOOBS card, 25–32
  - presentations in LibreOffice Impress, 134–136
  - programs, 241–255
  - scripts in Scratch, 178–179
  - sound samples, 411–413
  - times tables program, 212–221
  - variables, 197
- cropping photos in GIMP, 144–145
- Ctrl key, 71, 76, 123, 145, 450
- curly braces ({}), 105, 230
- currency sign, showing in LibreOffice Calc, 133
- current
  - defined, 300
  - value of, 3103
- current directory (.), 110, 111
- current sourcing, 311
- customizing
  - desktop environment, 82–83
  - with Linux commands, 122–124
- cutting files, 69

## D

- data
  - backing up, 84–85
  - in Scratch 2, 181
- data label, 56
- data loss, avoiding, 39
- data partition, setting up, 56
- Data partition option (NOOBS Lite), 48

- data sheet, 307
- data word, 366
- date command, 104
- DC (direct current), 302
- Debian Reference icon, 61
- debounce time, 336
- def statement, 231–232, 368
- Delete button (Image Viewer), 81
- deleting
  - directories in Linux Shell, 110–111
  - files and folders, 71
  - files in Linux Shell, 107–108
  - software, 117
  - sprites in Scratch, 184–185
- demo1ish() function, 270
- dependencies, 115, 456
- depth-first maze generation algorithm, 270
- desktop environment
  - about, 59–60
  - Applications menu, 60–63
  - changing displays for files, 72–73
  - closing program windows, 63–64
  - copying files/folders, 69
  - creating files/folders, 71
  - creating folders, 71
  - customizing, 82–83
  - File Manager, 65–73
  - icons, 59–60
  - Image Viewer, 79–81
  - Leafpad Text Editor, 81–82
  - moving files/folders, 69
  - navigating, 60–64
  - opening folders as root or in terminal, 73
  - playing music in, 158–159
  - Programs menu, 63
  - resizing program windows, 63–64
  - selecting files/folders, 70
  - starting, 59
  - Task Manager, 64–65
  - using external storage devices in, 67
  - web browsing, 74–78
- Desktop folder, 67
- desktop wallpaper, changing, 60
- detailed list view (File Manager), 72



- detecting
  - mouse clicks, 244–245
  - when sprites are hit, 196–197
- dev directory, 92
- devices, Bluetooth, 54
- DHCP (Dynamic Host Configuration Protocol), 46, 457
- dictionaries, 229–231, 231–236
- dictionarycheck() function, 232, 233, 236
- diffusers, 359–360
- Dillo browser, 74
- diode, 301
- direct current (DC), 302
- directions, moving sprites in Scratch with, 168–170
- directories
  - changing in Linux Shell, 89
  - creating in Linux Shell, 106
  - listing, 88–89
  - removing in Linux Shell, 110–111
- directory tree, 91–93
- disconnecting peripherals, 447
- display socket, 41–42
- displaying
  - currency sign in LibreOffice Calc, 133
  - sprite information on Stage, 172
- displayRoll function, 346
- displays, changing for files, 72–73
- distribution (distros), 24
- documents, saving as PDF, 131
- dollar sign (\$), 88
- DotStar, 362–365
- double ampersand (&&), 116
- double equal sign (==), 228
- double slash (//) operator, 211
- double-throw switch, 303
- downloading
  - NOOBS, 25
  - operating system, 23–33
- draw() function, 243, 246, 250, 251, 252
- drawing sprites in Scratch, 186–189
- drumbeat, synchronizing with in Sonic Pi, 293
- Dtronixs (website), 325
- Dummies (website), 4
- duplicate copy, 69
- duplicating sprites, 203

- DVI display, 42
- Dynamic Host Configuration Protocol (DHCP), 46, 457

## E

- Easel icon, 204
- eBay (website), 19
- echo command, 104, 122
- edge detection, 378
- Edit menu (Leafpad Text Editor), 81
- editing photos. *See* GIMP
- egg box, 360
- Electric Skateboard project, 439
- electricity, nature of, 300–306
- Element14 (website), 15
- Ellipse tool (Scratch), 187
- else command, 236
- else statement, 274
- email, sending and receiving with Claws Mail, 78
- embedded system, 308
- enabling
  - HDMI safe mode, 451
  - keyboard control, 192–193
  - multiple clouds to be clicked, 251–252
  - sprites to control sprites, 193–195
- enroll function, 399
- entering
  - commands, 103
  - Python commands, 208–210
- environmental regulations, 315
- Epiphany browser, 74
- equivalent circuit, 306
- Eraser tool (Scratch), 187
- escape sequence, 217
- etc directory, 92
- Ethernet connection, 457
- Ethernet socket, 45
- Events block, in Scratch 2, 181
- execute permission, 101
- Exit Image Viewer button (Image Viewer), 81
- external storage devices
  - about, 18
  - mounting, 455–456
  - using in desktop environment, 67
- Eyedropper tool (Scratch), 188

## F

- Farnell, 319
- Fernandez, David (website), 439
- file command, 89–90, 110
- File Manager
  - about, 65–66
  - changing file displays, 72–73
  - copying files/folders, 69
  - creating blank files, 71
  - creating folders, 71
  - moving files/folders, 69
  - navigating, 66–69
  - opening folders as roots or in terminals, 73
  - selecting files/folders, 70
- File menu, 128, 131
- File menu (Leafpad Text Editor), 81
- FileName function, 400
- files and folders
  - bookmarking, 67
  - changing displays for, 72–73
  - checking types in Linux Shell, 89–90
  - copying, 69, 111–113
  - copying in Linux Shell, 111–113
  - creating, 71, 104–105
  - creating using redirection in Linux Shell, 104–105
  - cutting, 69
  - defined, 65
  - deleting, 71, 107–108
  - deleting in Linux Shell, 107–108
  - listing, 88–89
  - moving, 69
  - naming, 105
  - opening, 65, 73
  - opening as root or in terminal, 73
  - renaming in Linux Shell, 111–113
  - selecting, 70, 108–110
  - selecting using wildcards, 108–110
  - setting up, 241
- Fill tool (Scratch), 187
- finding
  - applications, 83–84
  - package name, 114–115
- firing mechanism, 193–195
- firingflag variable, 201
- Fisheye effect (Scratch), 175
- Fit Image to Window button (Image Viewer), 79
- fixing
  - bugs, 201–203
  - software installation issues, 456
- flags, 201
- Flash, 74
- flashing SD cards, 32–33
- Flip Horizontally button (Image Viewer), 80
- Flip Vertically button (Image Viewer), 80
- flipping photos in GIMP, 145
- flux, 314–315
- FocusWriter, 429
- Folder History (File Manager), 69
- folders and files
  - bookmarking, 67
  - changing displays for, 72–73
  - checking types in Linux Shell, 89–90
  - copying, 69, 111–113
  - copying in Linux Shell, 111–113
  - creating, 71, 104–105
  - creating using redirection in Linux Shell, 104–105
  - cutting, 69
  - defined, 65
  - deleting, 71, 107–108
  - deleting in Linux Shell, 107–108
  - listing, 88–89
  - moving, 69
  - naming, 105
  - opening, 65, 73
  - opening as root or in terminal, 73
  - renaming in Linux Shell, 111–113
  - selecting, 70, 108–110
  - selecting using wildcards, 108–110
  - setting up, 241
- Fontwork text, 138
- footer, 366
- for loop, 218–221
- for statement, 268
- Forever block, 191–192
- format, long listing, 99–102

- Format menu (LibreOffice Writer), 131
- formatting
  - MicroSD card, 26–30
  - presentations in LibreOffice Impress, 134–136
  - SD card, 26–30
- foundations, laying in Minecraft, 266–267
- Fraqrive, 431–432
- free software, 24
- freeing up space, 117
- Full Screen button (Image Viewer), 79
- functions
  - animate(), 246
  - append(), 224, 271
  - calling, 236
  - checkRequest, 355
  - collidepoint(), 245
  - countdown(), 253
  - creating in Minecraft, 270–271
  - demolish(), 270
  - dictionarycheck(), 232, 233, 236
  - displayRoll, 346
  - draw(), 243, 246, 250, 251, 252
  - enroll, 399
  - FileName, 400
  - getString, 400
  - hideMaker(), 270
  - input(), 216, 226, 229
  - len(), 223
  - main, 345, 350, 378
  - pattern, 387
  - pause, 337
  - randCol, 388
  - random.randint(), 226, 247
  - range(), 219, 220
  - realx(), 270
  - realz(), 270
  - rrand\_i(), 289
  - screen.draw.text(), 250
  - showMaker(), 270
  - str(), 249
  - testAllWalls(), 270, 271
  - time.sleep(), 348
  - update(), 251

## G

- games, playing, 204
- General Purpose Input/Output (GPIO) pins
  - about, 308–309
  - on Raspberry Pi, 321–327
  - soldering, 314–316
  - soldering onto Pi Zero, 323–324
  - uses for, 309
  - using as inputs, 313–314
  - using output pins, 311–313
  - what they do, 309–311
- generating
  - chatbot program, 221–238
  - dictionary look-up function, 231–236
  - directories in Linux Shell, 106
  - files using redirection in Linux Shell, 104–105
  - files/folders, 71
  - folders, 71
  - functions in Minecraft, 270–271
  - invitations in LibreOffice Draw, 136–138
  - main conversation loop, 236
  - main loop in Minecraft, 271–274
  - NOOBS card, 25–32
  - presentations in LibreOffice Impress, 134–136
  - scripts in Scratch, 178–179
  - times tables program, 212–221
  - variables, 197
- getString function, 400
- Ghost effect (Scratch), 175
- GIMP (GNU Image Manipulation Program)
  - about, 139–140
  - adjusting colors, 145
  - converting images between formats, 147
  - cropping photos, 144–145
  - flipping photos, 145
  - installing, 140
  - repairing photos, 146–147
  - resizing images, 142–144
  - rotating photos, 145
  - screen layout, 140–142
  - starting, 140
  - website, 147

- Glide block, 195
- global variables, 252
- GNU Image Manipulation Program (GIMP)
  - about, 139–140
  - adjusting colors, 145
  - converting images between formats, 147
  - cropping photos, 144–145
  - flipping photos, 145
  - installing, 140
  - repairing photos, 146–147
  - resizing images, 142–144
  - rotating photos, 145
  - screen layout, 140–142
  - starting, 140
  - website, 147
- GNU Project, 24
- GNU/Linux. *See* Linux
- Go to Original Size button (Image Viewer), 79
- GParted, 26
- GPIO (General Purpose Input/Output) pins
  - about, 308–309
  - on Raspberry Pi, 321–327
  - soldering, 314–316
  - soldering onto Pi Zero, 323–324
  - uses for, 309
  - using as inputs, 313–314
  - using output pins, 311–313
  - what they do, 309–311
- GPIO Zero, creating LED flash using, 334–337
- graphic effects (Scratch), 175
- graphics, creating, 413–423
- green flag, 190–191
- grid coordinates, moving sprites in Scratch with, 170–172
- Grisbi, 433
- groups command, 118

## H

- hard drive, external, 18
- hash mark (#), 213, 450, 451
- HAT board, 317–319
- HDMI (high definition multimedia interface), 16
- HDMI cable, 42

- HDMI CEC (Consumer Electronics Council) standard, 158
- HDMI converter, 41
- HDMI display, 42
- HDMI safe mode, enabling, 451
- Heart Rate Monitor project, 439
- help facility, 121
- Hide block (Scratch), 176
- hideMaker() function, 270
- high definition multimedia interface (HDMI), 16
- #highaltitudechannel (website), 442
- History pane (GIMP), 141–142
- Home (File Manager), 69
- home directory, 92
- hyphen (-) operator, 88, 97, 100–101, 105, 211

## I

- icon bar (File Manager), 68–69
- icons
  - desktop, 50–60
  - Easel, 204
  - explained, 3–4
  - IDLE 3, 260
  - scissors, 184
- IDE (integrated development environment), 208
- IDLE (Python IDE), 208, 223, 260
- IDLE 3 icon, 260
- If block, 194
- if statement, 274, 330, 375
- image files, in LibreOffice Impress, 135
- Image Viewer, using, 79–81
- incompatible devices, 21
- indentations, 220
- index number, 229
- info page, 122
- input command, 220
- input() function, 216, 226, 229
- input/output pins, 309
- inputs, using GPOPs as, 313–314
- Insert menu (LibreOffice Writer), 131
- inserting
  - MicroSD card, 37–39
  - SD card, 37–39

- installing
  - applications, 83–84
  - Beneath a Steel Sky, 434
  - FocusWriter, 429
  - Fraqrive, 431
  - GIMP, 140
  - Grisbi, 433
  - LibreOffice, 128
  - multiple operating systems, 49
  - Penguins Puzzle, 428
  - software, 113–118, 115
  - Tux Paint, 432
  - XInvaders 3D, 431
- integrated development environment (IDE), 208
- Internet connection, troubleshooting, 446
- Internet resources
  - Adafruit, 15, 325
  - advanced settings, 454
  - Akerman, Dave, 9, 442
  - Bryan, David, 439
  - Cheat Sheet, 4
  - Clemens, Michael, 437
  - compatible devices, 21
  - Dtronixs, 325
  - Dummies, 4
  - eBay, 19
  - Electric Skateboard project, 439
  - Element14, 15
  - Fernandez, David, 439
  - FocusWriter, 429
  - GIMP (GNU Image Manipulation Program), 147
  - HAT boards, 317
  - #highaltitudechannel, 442
  - incompatible devices, 21
  - LibreOffice, 128
  - Minecraft Wiki, 263
  - online help, 448
  - photobooth, 440
  - Pi Hut, 15
  - Pibow, 19
  - Pimoroni, 15
  - Pope, Daniel (blog), 256
  - Pygame Zero built-in objects list, 256
  - Pygame Zero documentation, 256

- Raspberry Pi, 25, 156
- Raspberry Pi Camera Module, 54–55
- RS Components, 15
- Scratch, 205
- SD cards, 447
- SDFormatter, 26
- simulator, 307
- Tux Paint, 433
- UK High Altitude Society, 442
- updates, 4
- Wardell, Steve, 438
- Way Out West Hackathon, 440

## J

- Jackson, Michael (singer), 179
- Jukebox project, 395–400
- jumper wires, 326

## K

- Keepy Uppy, 375–378
- key, USB, 18
- keyboard
  - adjusting sensitivity, 83
  - connecting, 44
  - enabling control of, 192–193
  - USB, 17
- keyboard shortcuts, 70
- Kodi system, 150

## L

- Lakka option (NOOBS Lite), 48
- Layers pane (GIMP), 141–142
- Leafpad Text Editor, 81–82
- leaps *i*ze variable, 199–200, 203
- LED flash
  - about, 327–328
  - controlling flashing speed, 329–331
  - using GPIO Zero, 334–337
  - using Python, 331–334
  - using Scratch 1.4, 328–329
- LED matrix, 382–388
- LED strips, 382

- LEDs (light-emitting diodes)
  - about, 311–312, 378
  - current limits, 379
  - display update, 381
  - getting more, 381–388
  - memory, 379–380
  - signals, 379–380
- len() function, 223
- less command, 102–103, 121
- Less command, 102–103
- lib directory, 92
- LibreELEC, 150
- LibreELEC option (NOOBS Lite), 47
- LibreOffice icon, 62
  - about, 127
  - installing, 128
  - LibreOffice Calc, 131–134
  - LibreOffice Draw, 136–138
  - LibreOffice Impress, 134–136
  - LibreOffice Math, 129
  - LibreOffice Writer, 129–131
  - saving work in, 129
  - starting, 128–129
  - website, 128
- license() command, 209
- light-emitting diodes (LEDs)
  - about, 311–312, 378
  - current limits, 379
  - display update, 381
  - getting more, 381–388
  - memory, 379–380
  - signals, 379–380
- Lightweight X11 Desktop Environment (LXDE), 59
- line drawing, 382–388
- Line tool (Scratch), 187
- Linux
  - about, 14, 24
  - copying NOOBS to SD or MicroSD cards, 30–32
  - SD cards and, 28–30
- Linux commands
  - about, 120–122
  - canceling, 102
  - customizing with, 122–124
- Linux Foundation, 24
- Linux Shell
  - about, 87
  - absolute paths, 94–96
  - advanced listing options, 96–99
  - calculating sums with, 210–212
  - changing directories, 89
  - changing parent directory, 90–91
  - checking file types, 89–90
  - copying files, 111–113
  - creating directories, 106
  - creating files using redirection, 104–105
  - customizing with Linux commands, 122–124
  - deleting files, 107–108
  - directory tree, 91–93
  - entering commands, 103
  - installing software, 113–118
  - Less command, 102–103
  - Linux commands, 120–122
  - listing files/directories, 88–89
  - long listing format and permissions, 99–102
  - managing software, 113–118
  - managing user accounts, 118–120
  - naming files, 105
  - prompt, 88
  - rebooting, 124
  - relative paths, 94–96
  - removing directories, 110–111
  - renaming files, 111–113
  - selecting multiple files using wildcards, 108–110
  - shutting down, 124
- list names, using in programs, 288
- listing files/directories, 88–89
- lists
  - about, 222–225
  - creating random chat program using, 225–227
  - setting up in Minecraft, 269–270
- Lite version (NOOBS), 46
- live loops, 289–291
- local variable, 252
- Log, 289
- logging in, 50
- logging out, from PIXEL, 85

- logic levels, 310
- long listing, format and permissions, 99–102
- Looks button (Scratch), 173–174
- loops
  - creating Minecraft, 271–274
  - defined, 212
- lost+found directory, 92
- ls command, 97–99
- LXDE (Lightweight X11 Desktop Environment), 59

## M

- mA (milliamps), 306
- Mac
  - copying NOOBS to SD or MicroSD cards, 30
  - SDFormatter software, 27–28
- Mac Mini, 441
- Magic Mirror project, 440–441
- main function, 345, 350, 378
- making
  - chatbot program, 221–238
  - dictionary look-up function, 231–236
  - directories in Linux Shell, 106
  - files using redirection in Linux Shell, 104–105
  - files/folders, 71
  - folders, 71
  - functions in Minecraft, 270–271
  - invitations in LibreOffice Draw, 136–138
  - main conversation loop, 236
  - main loop in Minecraft, 271–274
  - NOOBS card, 25–32
  - presentations in LibreOffice Impress, 134–136
  - scripts in Scratch, 178–179
  - times tables program, 212–221
  - variables, 197
- managing
  - scripts, 190–195
  - software, 113–118
  - user accounts, 118–120
- Mathematica, 429–431
- Mathematica icon, 62
- Maximize button, 63
- maze algorithm (Minecraft), 268–269
- maze parameters, setting in Minecraft, 265–266
- maze walls, placing in Minecraft, 267–268
- McManus, Sean (author)
  - Cool Scratch projects in Easy Steps*, 205
  - Mission Python*, 256
  - Scratch Programming in Easy Steps*, 205
- mc.player.getTilePos() command, 279
- Mechanical Turk, 442
- media center
  - adding media, 151–155
  - adding music, 152–153
  - adding pictures, 154
  - adding to media center, 151–155
  - adding videos, 153–154
  - changing settings, 157–158
  - navigating, 150–151
  - playing music, 155–156
  - playing music in desktop environment, 158–159
  - playing videos, 156–157
  - remote control, 158
  - setting up, 149–150
  - streaming media, 155
  - turning off, 158
  - viewing photos, 157
- media directory, 92
- memory, 379–380
- menu bar
  - File Manager, 67
  - GIMP, 141
  - media center, 150–151
- Micro USB socket, 41
- MicroSD card
  - about, 17
  - checking, 447
  - copying NOOBS to, 30–32
  - formatting, 26–30
  - inserting, 37–39
  - replacing, 447
- MicroSD card writer, 17
- Microsoft Excel. *See* LibreOffice, LibreOffice Calc
- MIDI note numbers, 283–284

- MIFARE card, 393–395
- milliamps (mA), 306
- Minecraft
  - breaking blocks, 259–260
  - changing perspective in, 258
  - minecraft module, 261–278
  - moving around in, 258–259
  - playing, 258–260
  - preparing for Python, 260
  - starting, 258
- Minecraft Maze Maker code, 275–278
- minecraft module
  - about, 261
  - adding blocks, 263–264
  - adding ceilings, 274
  - coordinates, 261–262
  - creating functions, 270–271
  - creating main loop, 271–274
  - laying foundations, 266–267
  - maze algorithm, 268–269
  - placing maze walls, 267–268
  - positioning players, 274
  - repositioning players, 262–263
  - setting maze parameters, 265–266
  - setting up variables and lists, 269–270
  - stopping players from changing world, 264–265
- Minecraft Pi
  - about, 257–258
  - icon, 62
- Minecraft: Pi Edition*, 257–258
- minecraft.Minecraft.create(), 261
- Mini HDMI socket, 41
- Minimize button, 64
- Mission Python* (McManus), 256
- mkdir command, 106, 107, 121
- mnt directory, 93
- Model 2, inserting SD cards in, 37–39
- Model A
  - about, 13, 36
  - audio socket, 45
  - connecting to Internet, 53
  - inserting SD cards in, 37–39
  - network connections, 45
- Model A+
  - about, 12, 36
  - inserting SD cards in, 37–39
  - network connections, 45
- Model B
  - about, 12, 13, 36
  - audio socket, 45
  - Ethernet socket, 46
  - inserting SD cards in, 37–39
- Model B+, 13
  - about, 36
  - audio socket, 45
  - inserting SD cards in, 37–39
- Model Zero
  - about, 12
  - Ethernet socket, 46
  - inserting SD cards in, 38–39
  - network connections, 45
  - soldering GPIO pins onto, 323–324
- Model Zero W
  - about, 12, 37
  - connecting camera on, 39
  - inserting SD cards in, 38–39
- modifying
  - appearance of sprites in Scratch, 172–176
  - audiobook, 438
  - background in Scratch, 185
  - colors in GIMP, 145
  - desktop wallpaper, 60
  - directories in Linux Shell, 89
  - displays for files, 72–73
  - game speed, 204
  - parent directory in Linux Shell, 90–91
  - perspective in Minecraft, 258
  - screen display, 452–454
  - sensitivity of keyboard/mouse, 83
  - settings, 448–454
  - settings in media center, 157–158
  - sprite visibility in Scratch, 176
- modules, 226
- monitor
  - about, 16
  - connecting, 41–43
- Mosaic effect (Scratch), 175



- Motion block, 196
- mounting external storage devices, 455–456
- mouse
  - adjusting sensitivity, 83
  - connecting, 44
  - USB, 17
- mouse clicks, detecting, 244–245
- moving
  - around in Minecraft, 258–259
  - files/folders, 69
  - sprites automatically, 199–200
  - sprites in Scratch, 167–172
- MPEG2 format, 156
- music. *See also* Sonic Pi
  - adding, 152–153
  - adding in Scratch, 176–178
  - composing using `shuffle` method, 287
  - playing in desktop environment, 158–159
  - playing in media center, 155–156

## N

- naming
  - files, 105
  - sprites, 189
- Nano text editor, 449–451
- navigating
  - desktop environment, 60–64
  - File Manager, 66–69
  - media center, 150–151
- negative, 302
- NeoPixels, 362–365
- Netsurf browser, 74
- network connection, troubleshooting, 457
- networked media, adding to media center, 151–155
- New Bookmark window, 76
- new-out-of-box software. *See* NOOBS
- Next button (Image Viewer), 79
- Next Folder (File Manager), 69
- nonlinear device, 307
- NOOBS
  - copying to SD or MicroSD card, 30–32
  - downloading, 25
  - Lite version, 46

- NOOBS card
  - creating, 25–32
  - using, 32
- NOOBSLite, 49
- Normalize option (GIMP), 145
- note names (Sonic Pi), 285–286
- note numbers (Scratch), 176–178
- notes, playing in Sonic Pi, 283–285, 288–289
- `numberOfCells` variable, 269
- `numberOfVisitedCells` variable, 269
- numbers, printing, 216–218

## O

- ODF Text Document (.odt), 130
- offline and network install version, of NOOBS, 25
- .ogg sound format, 240
- ohms, 300
- Old McDonald's Farm, 410–423
- 13.56 MHz tags, 390
- 125-135 KHz tags, 390
- One-Button Audiobook Player project, 437–438
- online help, 448
- on/off switch, 46
- Open File button (Image Viewer), 80
- opening
  - `config.txt` file, 449
  - folders, 65
  - folders as root or in terminal, 73
- operating system
  - about, 23
  - creating NOOBS card, 25–32
  - installing multiple, 49
  - Linux, 24
  - using NOOBS card, 32
- `opt` directory, 93
- options, for `ls` command, 98–99
- Options menu (Leafpad Text Editor), 82
- OSMC option (NOOBS Lite), 47
- output impedance, 307
- outputs, using GPIO pins as, 311–313
- overclocking, 52

## P

- package manager, 113
- package name, finding, 114–115
- Paint Editor (Scratch), 186–189
- Paintbrush tool (Scratch), 187
- panes (GIMP), 141–142
- Panflute Hero project, 440
- paper dolls, 401–410
- parent directory (..), 90–91, 110, 111
- passive tags, 390
- passwd command, 120
- Path (File Manager), 69
- pattern function, 387
- pause function, 337
- PDF, saving documents as, 131
- Pedestrian Crossing project, 347–356
- Penguins Puzzle, 428
- Pennec, Mélanie (developer), 443
- percent (%) operator, 211
- peripherals, 447
- permissions, long listing, 99–102
- perspective, in Minecraft, 258
- photo editing. *See* GIMP
- photobooth, 440–441
- Photobot.Co, 440–441
- photos
  - converting between formats in GIMP, 147
  - cropping in GIMP, 144–145
  - flipping in GIMP, 145
  - repairing in GIMP, 146–147
  - rotating in GIMP, 145
  - viewing in media center, 157
- physical computing, 299
- pi, 88
- pi folder, 67
- Pi Hut (website), 15
- Pi in the Sky project, 441–442
- Pi Model 2, inserting SD cards in, 37–39
- Pi Model A
  - about, 13, 36
  - audio socket, 45
  - connecting to Internet, 53
  - inserting SD cards in, 37–39
  - network connections, 45
- Pi Model A+
  - about, 12, 36
  - inserting SD cards in, 37–39
  - network connections, 45
- Pi Model B
  - about, 12, 13, 36
  - audio socket, 45
  - Ethernet socket, 46
  - inserting SD cards in, 37–39
- Pi Model B+, 13
  - about, 36
  - audio socket, 45
  - inserting SD cards in, 37–39
- Pi Model Zero
  - about, 12
  - Ethernet socket, 46
  - inserting SD cards in, 38–39
  - network connections, 45
  - soldering GPIO pins onto, 323–324
- Pi Model Zero W
  - about, 12, 37
  - connecting camera on, 39
  - inserting SD cards in, 38–39
- Pibow (website), 19
- pictures, adding, 154
- pidel command, 123
- Pimoroni (website), 15
- ping command, 457
- pipe (|) character, 121
- PIXEL
  - about, 59–60
  - logging out from, 85
- Pixelate effect (Scratch), 175
- players
  - forcing replies from, 228–229
  - positioning in Minecraft, 274
  - repositioning in Minecraft, 262–263
  - stopping form changing world in Minecraft, 264–265
- playerx variable, 269
- playerz variable, 269
- playing
  - games, 204
  - Minecraft, 258–260

- music in desktop environment, 158–159
- music in media center, 155–156
- notes in Sonic Pi, 283–285
- random notes in Sonic Pi, 288–289
- videos in media center, 156–157
- playlists, 156
- plus (+) operator, 211
- Polar H7 heartbeat sensor, 439
- pop() method, 23
- Pope, Daniel (developer), 239
  - blog, 256
- positioning
  - blocks in Minecraft, 259
  - players in Minecraft, 274
  - sprites in Scratch, 166–167
- positive, 302
- power supply
  - about, 19
  - connecting, 46–50
  - replacing, 448
- Preferences button (Image Viewer), 81
- presentations. *See* LibreOffice, LibreOffice Impress
- Previous button (Image Viewer), 79
- Previous Folder (File Manager), 68–69
- print command, 209, 210–212, 213–215, 216–218, 219
- printing words, variables, and numbers, 216–218
- privacy, protecting, 77–78
- proc directory, 93
- programming. *See also* Scratch
  - about, 164
  - arcade game. *See* arcade game, programming
  - Minecraft with Python. *See* Minecraft in Python. *See* Python
- programs. *See also specific programs*
  - creating, 241–255
  - running, 63, 213–214, 241–255
  - stopping, 64–65
  - using list names in, 288
  - writing in Sonic Pi, 286
- Programs menu, 63
- projects, 437–443. *See also specific projects*
- prompt (Linux Shell), 88

- protecting privacy, 77–78
- protocols, 283
- pull-up/down, 314
- Pulse-Wodth Modulation (PWM), 360–361
- punctuation, 230
- PWM (Pulse-Wodth Modulation), 360–361
- Pygame/Pygame Zero, 239–240. *See also* Cloudbusting game
- Python. *See also* Cloudbusting game
  - about, 207–208
  - calculating sums with Shell, 210–212
  - creating chatbot program, 221–238
  - creating LED flash using, 331–334
  - creating times table program, 212–221
  - entering commands, 208–210
  - Minecraft. *See* Minecraft
  - starting Python, 207–208
- Python games icon, 62
- Python icons, 62

## Q

- quality (image), 144
- question mark (?) wildcard, 105, 109
- quotation (speech) marks ("), 105
- quotes, in Python, 223

## R

- Radio Frequency IDentification card (RFID)
  - about, 389
  - how it works, 390–401
  - Old McDonald's Farm, 410–423
  - paper dolls, 401–410
- Rainbow Invaders, 370–375
- randCo1 function, 388
- random module, 261
- random number seed, adjusting, 287–288
- random numbers
  - using in arcade game programming, 195–196
  - using in Cloudbuster game, 247–248
- random.randint() function, 226, 247
- range() function, 219, 220

- Raspberry Pi. *See also specific topics*
  - about, 9–11
  - accessories for, 16–21
  - getting, 15
  - GPIO, 321–327
  - projects. *See specific projects*
  - troubleshooting, 446–448
  - turning on, 46–50
  - upgrading software on, 116–117
  - uses for, 14–15
  - versions, 11–14
  - website, 25, 156
- Raspberry Pi Camera Module
  - about, 18
  - connecting, 39–41
  - testing, 54–55
- Raspberry Turk project, 442–443
- raspberrypi, 88
- Raspbian, 25
  - configuring Raspberry Pi in, 50–53
- Raspbian option (NOOBS Lite), 47
- Raspi-config, 448–449
- raspiutils, 54–55
- raspid, 55
- RCA cable, 43
- read permission, 101
- ready-made add-on boards, 317–320
- realx() function, 270
- realz() function, 270
- rebooting, 124
- Recalbox option (NOOBS Lite), 48
- receiving email with Claws Mail, 78
- Rectangle tool (Scratch), 187
- redirection, creating files using, 104–105
- Redon, Eric (developer), 443
- Reduction of Hazardous Substances (RoHS), 315
- regenerating clouds, 250–251
- relative paths, 94–96
- Reload button, 75
- Remember icon, 4
- remote control, 43, 158
- removing
  - directories in Linux Shell, 110–111
  - files in Linux Shell, 107–108
  - software, 117
  - sprites in Scratch, 184–185
- renaming files in Linux Shell, 111–113
- repairing photos in GIMP, 146–147
- replacing
  - cables, 447
  - MicroSD card, 447
  - peripherals, 447
  - power supply, 448
  - screen, 447
  - SD card, 447
- repositioning players in Minecraft, 262–263
- resistance, 300, 306
- resizing
  - columns in LibreOffice Calc, 133
  - images in GIMP, 142–144
  - panes in GIMP, 142
  - program windows, 63–64
  - sprites in Scratch, 166–167, 175–176
- resources, Internet
  - Adafruit, 15, 325
  - advanced settings, 454
  - Akerman, Dave, 9, 442
  - Bryan, David, 439
  - Cheat Sheet, 4
  - Clemens, Michael, 437
  - compatible devices, 21
  - Dtronixs, 325
  - Dummies, 4
  - eBay, 19
  - Electric Skateboard project, 439
  - Element14, 15
  - Fernandez, David, 439
  - FocusWriter, 429
  - GIMP (GNU Image Manipulation Program), 147
  - HAT boards, 317
  - #highaltitudechannel, 442
  - incompatible devices, 21
  - LibreOffice, 128
  - Minecraft Wiki, 263
  - online help, 448
  - photobooth, 440
  - Pi Hut, 15
  - Pibow, 19

- Pimoroni, 15
- Pope, Daniel (blog), 256
- Pygame Zero built-in objects list, 256
- Pygame Zero ducmmentation, 256
- Raspberry Pi, 25, 156
- Raspberry Pi Camera Module, 54–55
- RS Components, 15
- Scratch, 205
- SD cards, 447
- SDFormatter, 26
- simulator, 307
- Tux Paint, 433
- UK High Altitude Society, 442
- updates, 4
- Wardell, Steve, 438
- Way Out West Hackathon, 440
- restarting, 446
- RFID (Radio Frequency IDentification card)
  - about, 389
  - how it works, 390–401
  - Old McDonald's Farm, 410–423
  - paper dolls, 401–410
- RGB LEDs
  - about, 357–359
  - creating colors, 359–362
- ribbon cable, 317
- right-clicking folders, 69
- RISC OS option (NOOBS Lite), 47
- rm command, 107
- rmdir command, 110
- RoHS (Reduction of Hazardous Substances), 315
- root, opening folders as, 73
- root directory, 93
- Rotate Left button (Image Viewer), 80
- Rotate Right button (Image Viewer), 80
- rotating photos in GIMP, 145
- router, connecting, 45–46
- rand\_i() function, 289
- RS Components (website), 15
- run directory, 93
- running
  - programs, 63, 213–214, 241–255
  - software, 116

## S

- samples
  - defined, 291–292
  - using in Sonic Pi, 291–292
- Saugen, Lucas (developer), 439–440
- Save File As button (Image Viewer), 80
- Save File button (Image Viewer), 80
- saving
  - documents as PDF, 131
  - work in LibreOffice, 129
  - work in Scratch, 180–181
- sbin directory, 93
- scissors icon, 184
- Scratch. *See also* arcade game, programming
  - about, 163
  - adding sounds and music, 176–178
  - changing appearance of sprites, 172–176
  - creating scripts, 178–179
  - drawing sprites in, 186–189
  - moving sprites, 167–172
  - positioning sprites, 166–167
  - resizing sprites, 166–167
  - saving work in, 180–181
  - screen layout, 165–166
  - starting, 165
  - tools, 187–188
  - using wait block to slow down sprites, 180
  - website, 205
- Scratch 1.4, 328–329
- Scratch 2, new features in, 181
- Scratch icon, 62
- Scratch Programming in Easy Steps* (McManus), 205
- screen, replacing, 447
- screen display
  - adjusting, 452–454
  - troubleshooting, 452–453
- screen layout
  - Scratch, 165–166
  - Sonic Pi, 282–283
- screen.draw.text() function, 250
- Screenly Open Source Edition (OSE) option (NOOBS Lite), 49

- scripts
  - adding to Stage, 203
  - Control block, 190–191
  - controlling, 190–195
  - creating in Scratch, 178–179
  - enabling keyboard control, 192–193
  - enabling sprites to control sprites, 193–195
  - Forever block, 191–192
  - green flag, 190–191
- Scripts Area (Scratch), 165–166
- scrollbar, 75
- SD Association, 26
- SD card writer, 17
- SD cards
  - about, 17
  - checking, 447
  - copying NOOBS to, 30–32
  - flashing, 30–32
  - formatting, 26–30
  - inserting, 37–39
  - Linux and, 28–30
  - replacing, 447
- SDFormatter software
  - for Mac, 27–28
  - website, 26
  - for Windows, 26–27
- Search menu (Leafpad Text Editor), 82
- searching, within web pages, 76
- Secure Shell (SSH), 458
- Select tool (Scratch), 188
- selecting
  - files/folders, 70, 108–110
  - multiple files using wildcards, 108–110
- sending email with Claws Mail, 78
- Sense HAT, 318
- Sense HAT Emulator, 435
- Sense HAT emulator icon, 62
- Sensing block, 193, 197, 203
- sensitivity, adjusting, 83
- series circuit, 304
- settings
  - adjusting, 448–454
  - advanced, 454
  - changing in media center, 157–158
  - maze parameters in Minecraft, 265–266
- setup
  - data partition, 56
  - folders, 241
  - lists in Minecraft, 269–270
  - media center, 149–150
  - variables in Minecraft, 269–270
- Shell. *See* Linux Shell
- shift operator (<<), 366
- Show block (Scratch), 176
- showing
  - currency sign in LibreOffice Calc, 133
  - sprite information on Stage, 172
- showMaker ( ) function, 270
- shuffle method, 287, 289
- Shutdown icon, 62
- shutting down, 85, 124
- signals, 379–380
- simulators, testing circuits with, 307
- SIZE, 265
- Skywriter HAT, 318–319
- slash (/) operator, 105, 211
- sleep command, 284, 291–292
- smart playlists, 156
- sockets, 41
- software
  - installing, 113–118, 115
  - managing, 113–118
  - recommended, 427–436
  - removing, 117
  - running, 116
  - troubleshooting, 456
  - updating, 448
  - upgrading on Raspberry Pi, 116–117
- soldering
  - about, 299
  - General Purpose Input/Output (GPIO) pins, 314–316
  - GPIO pins onto Pi Zero, 323–324
- solitaire package name, 115
- Sonic Pi
  - about, 281–282
  - adding special effects, 292
  - changing random number seed, 287–288

- composing random tunes using `shuffle`, 287
- live loops, 289–291
- playing notes, 283–285
- playing random notes, 288–289
- screen layout, 282–283
- synchronizing with drumbeat, 293
- using chord names, 285–286
- using list names in programs, 288
- using note names, 285–286
- using samples, 291–292
- writing programs, 286
- Sonic Pi icon, 62
- Sound Fighter project, 443
- sounds
  - adding in Scratch, 176–178
  - creating samples, 411–413
- space, freeing up, 117
- speakers, 19
- special effects, adding in Sonic Pi, 292
- speech bubbles (Scratch), 173–174
- speed, adjusting for game, 204
- `split()` method, 234
- Spotify, 440
- spreadsheet, 131–134
- Sprite List (Scratch), 165–166
- sprites
  - adding to games, 185–186
  - changing appearance of in Scratch, 172–176
  - changing visibility of in Scratch, 176
  - deleting in Scratch, 184–185
  - detecting when hit, 196–197
  - drawing in Scratch, 186–189
  - duplicating, 203
  - enabling to control sprites, 193–195
  - moving automatically, 199–200
  - moving in Scratch, 167–172
  - naming, 189
  - positioning in Scratch, 166–167
  - resizing in Scratch, 166–167, 175–176
  - slowing down with wait block, 180
- square brackets (`[]`), 105, 245
- `srv` directory, 93
- SSH (Secure Shell), 458
- Stage (Scratch), 165–166, 172, 203
- Stallman, Richard (developer), 24
- Stamp tool (Scratch), 188
- Start Slideshow button (Image Viewer), 79
- starting
  - desktop environment, 59
  - File Manager, 65
  - FocusWriter, 429
  - GIMP, 140
  - LibreOffice, 128–129
  - Mathematica, 429
  - Minecraft, 258
  - Penguins Puzzle from shell, 428
  - projects in Scratch, 184–185
  - Python, 207–208
  - Scratch, 165
  - Tux Paint, 432
- state machine, 348
- statements, conditional, 235. *See also specific statements*
- stopping
  - players from changing worlds in Minecraft, 264–265
  - programs, 64–65
- `str()` function, 249
- streaming media, using in media center, 155
- string methods, 234
- SUM formula (LibreOffice Calc), 134
- sums, calculating with Python Shell, 210–212
- switches, 302–305
- switching between tabs, 76
- switching off Raspberry Pi, 151
- switching on Raspberry Pi, 124
- `sync` command, 293
- synchronizing, with drumbeat in Sonic Pi, 293
- `sys` directory, 93, 261

## T

- tabbed browsing, 76
- `tablenum` variable, 216–218, 218–221
- tack switch, 328–329
- tags, 390
- task bar, 178
- Task Manager, 64–65

- taskbar, 60
- Technical Stuff icon, 4
- terminal, opening folders in, 73
- Terminal icon, 63
- testAllWalls() function, 270, 271
- testing
  - circuits with simulators, 307
  - Raspberry Pi Camera Module, 54–55
- Text tool (Scratch), 188
- Thonny Python IDE, 332
- thought bubbles (Scratch), 173–174
- thumbnail view (File Manager), 72
- tilde (~) symbol, 90–91, 94–95
- timers, adding, 253
- times tables program
  - about, 212
  - accepting user input, 216
  - creating, 212–221
  - printing words, variables, and numbers, 216–218
  - using for loops, 218–221
  - using variables, 214–215
- time.sleep() function, 348
- Tip icon, 3
- TLXOS option (NOOBS Lite), 49
- tmp directory, 93
- tools (Scratch), 187–188
- Tools menu, 73
- Torvalds, Linus (developer), 24
- troubleshooting Raspberry Pi, 446–448
- T-Shirt Cannon project, 439–440
- tuple, 244
- turning off media center, 158
- turning on, 46–50
- Tux Paint, 432–433
- TV
  - about, 16
  - connecting, 41–43
- .txt, 104, 110
- type command, 120

## U

- Ubuntu, 28–30
- UHF (860-960MHz), 390
- UID (Unique IDentification number), 393–394

- UK High Altitude Society (website), 442
- Undelete option (Scratch), 184
- Unique IDentification number (UID), 393–394
- Up a Level (File Manager), 69
- update() function, 251
- updating
  - cache, 114
  - Penguins Puzzle, 428
  - software, 448
  - website for, 4
- upgrading software on Raspberry Pi, 116–117
- Upton, Eben (designer), 11
- USB device, adding media to media center from, 151–152
- USB hub
  - about, 16–17
  - connecting, 43–44
- USB key, 18
- USB keyboard/mouse, 17
- USB socket, 43–44
- USB Wi-Fi adapter, 18
- user accounts, managing, 118–120
- user input, accepting, 216
- useradd command, 119
- users, adding, 119
- usr directory, 93

## V

- value of current, 310
- van Loo, Gert (designer), 16
- var directory, 93
- variables
  - about, 197–199
  - cellsVisitedList[], 269
  - creating, 197
  - firingflag, 201
  - global, 252
  - leapsize, 199–200, 203
  - local, 252
  - numberOfCells, 269
  - numberOfVisitedCells, 269
  - playerx, 269
  - playerz, 269
  - printing, 216–218



- setting up in Minecraft, 269–270
- tablenum, 216–218, 218–221
- using, 214–215
- xposition, 269
- zposition, 269
- Variables button (Scratch), 201
- VC1 format, 156
- verifying
  - connections, 446
  - file types in Linux Shell, 89–90
  - MicroSD card, 47, 447
  - SD card, 447
  - what's installed, 118
- versions
  - Python, 208
  - Raspberry Pi, 11–14
- VGA (video graphics array), 16
- video, 153–154, 156–157. *See also* media center
- video graphics array (VGA), 16
- Videocore IV graphics processing unit (GPU), 14
- View menu (File Manager), 72
- viewing photos in media center, 17
- visibility, changing of sprites in Scratch, 176
- VLC Media Player, 159
- voltage, 300
- voltage drop, 312

## W

- waffle box, 360
- wait block, slowing down sprites with, 180
- wallpaper, 60
- wallpaper, changing, 60
- Wardell, Steve (developer), 438
- Warning! icon, 4
- Way Out West Hackathon, 440
- Weather Station project, 438
- web browsing
  - about, 74
  - bookmarks, 76–77
  - with Chromium, 74–76
  - privacy, 77–78
  - searching within web pages, 76
  - tabbed browsing, 76
  - web pages, searching within, 76
- websites
  - Adafruit, 15, 325
  - advanced settings, 454
  - Akerman, Dave, 9, 442
  - Bryan, David, 439
  - Cheat Sheet, 4
  - Clemens, Michael, 437
  - compatible devices, 21
  - Dtronixs, 325
  - Dummies, 4
  - eBay, 19
  - Electric Skateboard project, 439
  - Element14, 15
  - Fernandez, David, 439
  - FocusWriter, 429
  - GIMP (GNU Image Manipulation Program), 147
  - HAT boards, 317
  - #highaltitudechannel, 442
  - incompatible devices, 21
  - LibreOffice, 128
  - Minecraft Wiki, 263
  - online help, 448
  - photobooth, 440
  - Pi Hut, 15
  - Pibow, 19
  - Pimoroni, 15
  - Pope, Daniel (blog), 256
  - Pygame Zero built-in objects list, 256
  - Pygame Zero ducmmentation, 256
  - Raspberry Pi, 25, 156
  - Raspberry Pi Camera Module, 54–55
  - RS Components, 15
  - Scratch, 205
  - SD cards, 447
  - SDFormatter, 26
  - simulator, 307
  - Tux Paint, 433
  - UK High Altitude Society, 442
  - updates, 4
  - Wardell, Steve, 438
  - Way Out West Hackathon, 440
  - Weessa, Jean (developer), 443
  - which command, 120

- while command, 229
- while loop, 227–228, 274, 350
- Whirl effect (Scratch), 175
- Wi-Fi, configuring, 53
- Wi-Fi adapter, 18
- wildcards, selecting files using, 108–110
- windows
  - closing, 63–64
  - resizing, 63–64
- Windows 10 IoT Core option (NOOBS Lite), 49
- Windows PC
  - copying NOOBS to SD or MicroSD cards, 30
  - SDFormatter software, 26–27
- Wolfram icon, 63
- words, printing, 216–218
- workspaces (Sonic Pi), 282–283
- write permission, 101
- writing
  - programs in Python. *See* Python
  - programs in Sonic Pi, 286
- WS2350 weather station, 438
- WS2812b, 362–365

## X

- X button, 63
- X server, 116
- XInvaders 3D, 431
- xposition variable, 269
- Xtrinsic Sense, 319–320

## Z

- Zero
  - about, 12
  - Ethernet socket, 46
  - inserting SD cards in, 38–39
  - network connections, 45
  - soldering GPIO pins onto, 323–324
- Zero W
  - about, 12, 37
  - connecting camera on, 39
  - inserting SD cards in, 38–39
- Zoom In button (Image Viewer), 79
- Zoom Out button (Image Viewer), 79
- zposition variable, 269

## About the Authors

**Sean McManus** is an expert technology and business author. His other books include *Mission Python*, *Coder Academy*, *Cool Scratch Projects in Easy Steps*, *Scratch Programming in Easy Steps*, and *Web Design in Easy Steps*. His novel for adults, *Earworm*, goes undercover in the music industry, exposing a conspiracy to replace bands with computer-generated music. His tutorials and articles have appeared in magazines including *The MagPi*, *Internet Magazine*, *Internet Works*, *Business 2.0*, *Making Music*, and *Personal Computer World*. He has been a Code Club volunteer, helping children at a local school to learn computer programming. Visit his website at [www.sean.co.uk](http://www.sean.co.uk) for bonus content from his books.

**Mike Cook** has been making electronic things since he was at school. A former lecturer in physics at Manchester Metropolitan University, he wrote more than 300 computing and electronics articles in the pages of computer magazines for 20 years starting in the 1980s. Leaving the University after 21 years when the physics department closed down, he got a series of proper jobs where he designed digital TV set-top boxes and access control systems. His other books include *Raspberry Pi Projects*, *Raspberry Pi Projects for Dummies*, and *Arduino Music and Audio Projects*. He also works with Drake Music Labs North, a charity for disabled musicians, developing accessible music equipment.

Now retired and freelancing, he spends his days surrounded by wires, patrolling the forums as Grumpy Mike.

## Dedication

To my wife, Karen, with thanks for all her support throughout this project and always. And to Leo, our wonderful son. —Sean

To my wife, Wendy, who always acts delighted whenever I show her yet another blinking LED. And also to the late Leicester Taylor, World War II radar researcher and inspirational supervisor of my post-graduate research at the University of Salford. —Mike

# Author's Acknowledgments

Thank you to my co-author, Mike, for bringing his electronics expertise and fantastic project ideas. Thank you to Craig Smith for commissioning us to write the first edition of this book; and to Katie Mohr, our acquisitions editor on the second and third editions. Thanks also to Linda Morris for her editing support on the first two editions; and to Paul Levesque, our project editor, and Becky Whitney, our copy editor on this third edition. Our technical editors Jason E Geistweidt (3rd edition), Ryan Walmsley (2nd edition), and Paul Hallett (1st edition) cast a careful eye over the text and code and made much appreciated suggestions. Olivier Engler, who translated the first edition into French, provided helpful feedback too. Thanks also to Lorna Mein and Natasha Lee in marketing, and to the . . . *For Dummies* team for making it all happen.

Many people helped with research or permissions requests, including Karen McManus, Sam Aaron, Eben Upton, Liz Upton, Leo McHugh, Mark Turner, Peter Sayer, John Hartnup, Bill Kendrick, Simon Cox, Jon Williamson, Paul Beech, Peter de Rivaz, Michał Męciński, Ruairi Glynn, Stephen Reville, Lawrence James, Bram Stolk, Adam Kemeny, Will Jessop, and David Bryan. We wouldn't have a book to write if it weren't for the wonderful work of the Raspberry Pi Foundation, the manufacturers who took a gamble on it, and the many thousands of people who have contributed to the Raspberry Pi's software. —Sean

I would like to thank Sean McManus for inviting me to contribute to this book and the staff at Wiley for making the process of producing this book as painless as possible. —Mike



## **Publisher's Acknowledgments**

**Acquisitions Editor:** Katie Mohr

**Project Editor:** Paul Levesque

**Copy Editor:** Becky Whitney

**Technical Editor:** Jason Geistweidt

**Editorial Assistant:** Owen Kaelble

**Senior Editorial Assistant:** Cherie Case

**Production Editor:** Magesh Elangovan

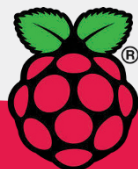
**Cover Image:** © Courtesy of Raspberry Pi  
Foundation

# Computing is for everyone



**Find out more about the Raspberry Pi  
Foundation, and how you can get involved  
in our community, at**

**[raspberrypi.org](https://raspberrypi.org)**



**Raspberry Pi**

The Raspberry Pi Foundation UK registered charity 1129409

# **WILEY END USER LICENSE AGREEMENT**

Go to [www.wiley.com/go/eula](http://www.wiley.com/go/eula) to access Wiley's ebook EULA.