

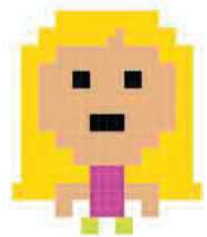
0 x =  } 0 1 0 :) 1 0(y = 0 * > 0 1 : 1 @ [/ 1 @ 1  0 : < / 0 * 0 1 # %**HELP YOUR KIDS WITH**

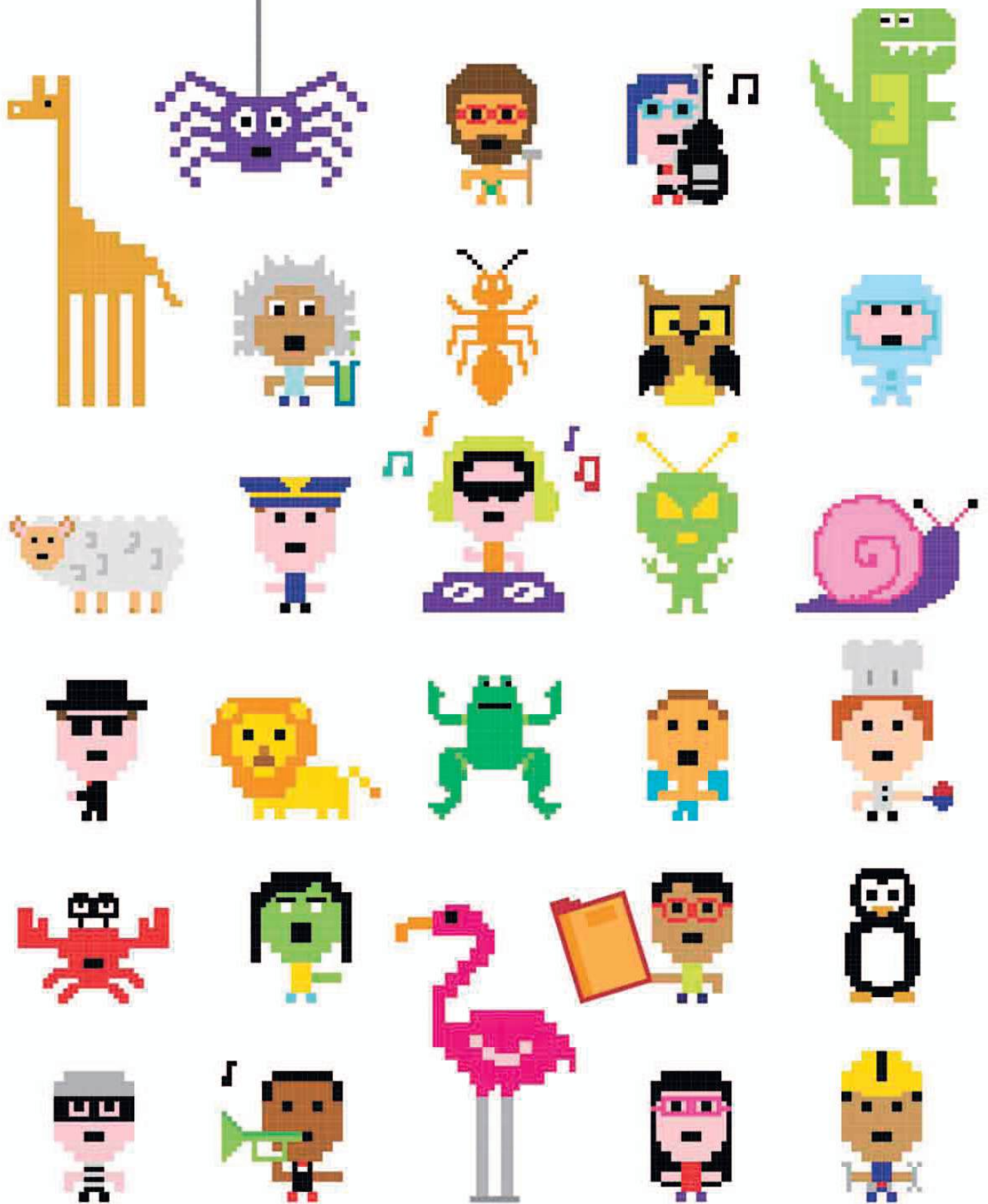
COMPUTER

CODING

0  1 : / y # 1 { 0 1 * <]1 0 : 1 > 1) @ x #  (1 0 } 1  0 \ = * 0 : 1 0 # @A UNIQUE STEP-BY-STEP VISUAL GUIDE,
FROM BINARY CODE TO BUILDING GAMES

HELP YOUR KIDS WITH
COMPUTer
CODing







HELP YOUR KIDS WITH
COMPUTer
CODing

A UNIQUE STEP-BY-STEP VISUAL GUIDE,
FROM BINARY CODE TO BUILDING GAMES





LONDON, NEW YORK, MELBOURNE,
MUNICH, AND DELHI

DK LONDON

Editor Sam Priddy
Designer Fiona Macdonald
Additional editors Sam Atkinson,
Lizzie Davey, Daniel Mills, Ben Morgan
US editor Jill Hamilton
Additional designer Simon Murrell
Managing editor Paula Regan
Managing art editor Owen Peyton Jones
Senior producer, pre-production Ben Marcus
Senior producer Mary Slater
Jacket editor Maud Whatley
Jacket designer Laura Brim
Jacket design development manager Sophia MTT
Publisher Sarah Larter
Art director Phil Ormerod
Associate publishing director Liz Wheeler
Publishing director Jonathan Metcalf

DK INDIA

Senior art editor Devika Dwarkadas
Editors Suefa Lee, Neha Pande
Art editors Sanjay Chauhan,
Shreya Anand Virmani
Assistant art editor Vanya Mittal
DTP designer Sachin Gupta
Managing editor Rohan Sinha
Deputy managing art editor Sudakshina Basu
Pre-production manager Balwant Singh
Jacket designer Suhita Dharamjit
Senior DTP designer Harish Aggarwal

First American Edition, 2014
Published in the United States by DK Publishing
345 Hudson Street, 4th Floor, New York, New York 10014
14 15 16 17 18 10 9 8 7 6 5 4 3 2 1
001 – 192672 – Jun/2014

Copyright © 2014 Dorling Kindersley Limited
All rights reserved. Without limiting the rights under copyright reserved above, 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, or otherwise), without the prior written permission of the copyright owner.
Published in Great Britain by Dorling Kindersley Limited.

A catalog record for this book is available from the Library of Congress.
ISBN: 978-1-4654-1956-9

DK books are available at special discounts when purchased in bulk for sales promotions, premiums, fund-raising, or educational use. For details, contact: DK Publishing Special Markets, 345 Hudson Street, New York, New York 10014 or SpecialSales@dk.com.

Printed and bound in China by South China Printing Company.
Discover more at
www.dk.com



CAROL VORDERMAN M.A. (CANTAB), MBE is one of Britain's best-loved TV presenters and is renowned for her skills in mathematics. She has a degree in Engineering from the University of Cambridge. Carol has a keen interest in coding, and feels strongly that every child should have the chance to learn such a valuable skill. She has hosted numerous TV shows on science and technology, such as *Tomorrow's World* and *How 2*, as well as *The Pride of Britain Awards*, on the BBC, ITV, and Channel 4. Whether co-hosting Channel 4's *Countdown* for 26 years, becoming the second best selling female nonfiction author of the noughties decade in the UK, or advising British Prime Minister David Cameron on the future of potential mathematics education in the UK, Carol has a passion and devotion to explaining mathematics, science, and technology in an exciting and easily understandable way.



DR. JON WOODCOCK M.A. (OXON) has a degree in Physics from the University of Oxford and a Ph.D. in Computational Astrophysics from the University of London. He started coding at the age of eight and has programmed all kinds of computers from single-chip microcontrollers to world-class supercomputers. His many projects include giant space simulations, research in high-tech companies, and intelligent robots made from junk. Jon has a passion for science and technology education, giving talks on space and running computer programming clubs in schools. He has worked on numerous science and technology books as a contributor and consultant.



SEAN McMANUS learned to program when he was nine. His first programming language was Logo. Today he is an expert technology author and journalist. His other books include *Scratch Programming in Easy Steps*, *Web Design in Easy Steps*, and *Raspberry Pi For Dummies*. Visit his website at www.sean.co.uk for Scratch games and tutorials.



CRAIG STEELE is a specialist in Computing Science education. He is Project Manager for CoderDojo Scotland, which runs free coding clubs for young people. Craig has previously worked for the Scottish Qualification Authority, Glasgow Science Centre, and the University of Glasgow. Craig's first computer was a ZX Spectrum.



CLAIRE QUIGLEY studied Computing Science at Glasgow University where she obtained a B.S. and a Ph.D. She has worked in the Computer Laboratory at Cambridge University and on a project that aimed to develop computational thinking skills in primary school pupils. She is a mentor at CoderDojo Scotland, a coding club for young people.



DANIEL McCAFFERTY holds a degree in Computer Science from the University of Strathclyde. Since graduating, he has been developing software for some of the world's largest investment banks. In his spare time, Daniel is a mentor at CoderDojo Scotland, a coding club for young people.

Contents

- 8 **FOREWORD** by Carol Vorderman
- 10 **HOW THIS BOOK WORKS**

1 WHAT IS CODING?

- 14 What is a computer program?
- 16 Thinking like a computer
- 18 Becoming a coder

2 STARTING FROM SCRATCH

- 22 What is Scratch?
- 24 Installing Scratch
- 26 Scratch interface
- 28 Sprites
- 30 Colored blocks and scripts
- 32 **Project 1: Escape the dragon!**
- 38 Making things move
- 40 Costumes
- 42 Hide and seek
- 44 Events
- 46 Simple loops
- 48 Pens and turtles
- 50 Variables
- 52 Math
- 54 Strings and lists
- 56 Coordinates
- 58 Make some noise
- 60 **Project 2: Roll the dice**
- 62 True or false?

- 64 Decisions and branches
- 66 Sensing and detecting
- 68 Complex loops
- 70 Sending messages
- 72 Creating blocks
- 74 **Project 3: Monkey mayhem**
- 82 Time to experiment

3 PLAYING WITH PYTHON

- 86 What is Python?
- 88 Installing Python
- 92 Introducing IDLE
- 94 Errors
- 96 **Project 4: Ghost game**
- 98 Ghost game decoded
- 100 Program flow
- 102 Simple commands
- 104 Harder commands
- 106 Which window?
- 108 Variables in Python
- 110 Types of data
- 112 Math in Python
- 114 Strings in Python
- 116 Input and output
- 118 Making decisions
- 120 Branching

- 122 Loops in Python
- 124 While loops
- 126 Escaping loops
- 128 Lists
- 130 Functions
- 132 **Project 5: Silly sentences**
- 134 Tuples and dictionaries
- 136 Lists in variables
- 138 Variables and functions
- 140 **Project 6: Drawing machine**
- 148 Bugs and debugging
- 150 Algorithms
- 152 Libraries
- 154 Making windows
- 156 Color and coordinates
- 158 Making shapes
- 160 Changing things
- 162 Reacting to events
- 164 **Project 7: Bubble blaster**
- 176 What next?

4 INSIDE COMPUTERS

- 180 Inside a computer
- 182 Binary and bases
- 184 Symbols and codes
- 186 Logic gates

- 188 Processors and memory
- 190 Essential programs
- 192 Storing data in files
- 194 The Internet

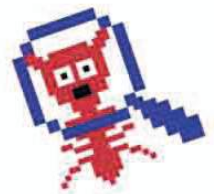
5 PROGRAMMING IN THE REAL WORLD

- 198 Computer languages
- 200 Coding stars
- 202 Busy programs
- 204 Computer games
- 206 Making apps
- 208 Programming for the Internet
- 210 Using JavaScript
- 212 Bad programs
- 214 Mini computers
- 216 Becoming a master programmer

- 218 Glossary
- 220 Index
- 224 Acknowledgments

Find out more at:

www.dk.com/computercoding



Foreword

Just a few years ago, computer coding seemed like a mysterious skill that could only be practiced by specialists. To many people, the idea that coding could be fun was a strange one. But then the world changed. In the space of a few years, the Internet, email, social networks, smartphones, and apps hit us like a tornado, transforming the way we live.

Computers are a huge part of life that we all now take for granted. Instead of calling someone on the phone, we send a text message or use social media. From shopping and entertainment to news and games, we guzzle everything computers have to offer. But we can do more than just use this technology, we can create it. If we can learn to code, we can make our own digital masterpieces.

Everything computers do is controlled by lines of code that someone has typed out on a keyboard. It might look like a foreign language, but it's a language anybody can pick up quite quickly. Many would argue that coding has become one of the most important skills you can learn in the 21st century.

Learning to code is tremendous fun because you can get instant results, no matter how much more you have to learn. In fact, it's such fun creating games and programs that it feels effortless once you're hooked. It's also creative—perhaps the first science that combines art, logic, storytelling, and business.

Not only that, coding is a fantastic skill for life. It strengthens logical thinking and problem-solving skills—vital in many different areas of life, from science and engineering to medicine and law. The number of jobs that require coding is set to increase dramatically in the future, and there's already a shortage of good coders. Learn to code, and the digital world is yours for the taking!

Carol Vorderman

CAROL VORDERMAN



How this book works

This book introduces all the essential concepts needed to understand computer coding. Fun projects throughout put these ideas into practice. Everything is broken down into small chunks so that it's easy to follow and understand.

Each topic is described in detail, with examples and exercises

"See also" boxes list other subjects that are linked to the topic



170 PLAYING WITH PY

BUBBLE BLASTER

Figuring out the distance
In this game, and lots of others, it is between two objects. Here's how to formula to have the computer work

11 This function calculates the distance between two objects. Add this bit of code directly to the code you wrote in step 9.

```
from math import sqrt
def distance(id1, id2):
    x1, y1 = get_coords(id1)
    x2, y2 = get_coords(id2)
    return sqrt((x2 - x1)
```

42 STARTING FROM SCRATCH

Hide and seek

Welcome to the special effects studio! Using the purple "Looks" blocks, find out how to make sprites vanish and reappear, grow and shrink, and fade in and out.

Hiding sprites
To make a sprite disappear, use the "hide" block. The sprite is still on the stage, and it can still move around, but it can't be seen unless the "show" block is used to make it visible again.

Hide and show
To make a sprite vanish, use the "hide" block. When you're ready for it to be seen again, use the "show" block. These blocks are found in the "Looks" section of the blocks palette.

Disappearing cat
Try this script using the cat sprite. It disappears and reappears but it keeps moving, even when you can't see it.

SEE ALSO
(38-39 Making things move
Sending 70-71) messages

Use the "hide" block to make sprites disappear in games

EXPERT TIPS
Showing sprites
Select a sprite in the sprite list. Click the "I" button on it to open the information panel. There you can also use the "show" tick box to show or hide a sprite.

Colorful illustrations highlight different programming concepts

Programming scripts and code are explained line by line

Instructions show what to click, drag, or select

Sizes and effects

Scripts can be used to change the size of a sprite and add special effects to it.

change size by 10
Type in positive numbers to make sprites bigger and negative numbers to make them smaller

set size to 100 %
Higher numbers make sprites bigger and lower numbers make them smaller. 100 is normal size

Changing a sprite's size
These two blocks can be used to make a sprite bigger or smaller, either by a set amount or by a percentage of its size.

Resets all the effects
The graph used to control or distort an experiment

Add
The graph used to control or distort an experiment

Using effects to teleport

Add a ghost sprite from the "Fantasy" category of the sprite library, and create the script shown below. It makes the ghost appear to teleport when clicked.

when this sprite clicked

clear graphic effects

repeat (20)
change ghost effect by (5)

glide (0.1) secs to x: pick random (-150) to (150) y: pick random (-150) to (150)

repeat (20)
change ghost effect by (-5)

The "ghost" effect makes the fade slightly; by repeating the block 20 times the sprite fades away completely

This "Operate" selects a random horizontal

Using this block makes the sprite fade back in

Labels help explain each step

Seven projects build up coding skills. Project pages are highlighted with a blue band

Simple step-by-step instructions guide you through each project

PYTHON BUBBLE BLASTER 171

Distance between points

Useful to know the distance between two points. Use a well-known mathematical formula to work it out.

Get the position of the first object
Get the position of the second object
Gives back the distance between them
 $\sqrt{(x2 - x1)^2 + (y2 - y1)^2}$

HIDE AND SEEK 43

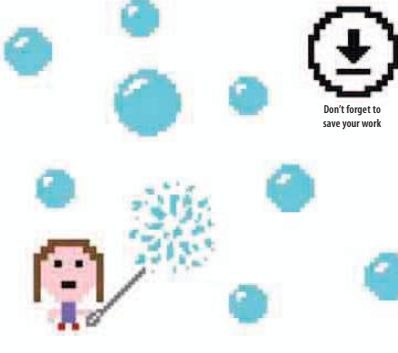
Change the numbers in the blocks to set how strong the effect is
pixelate effect by 25
effect to 0
Each color is represented by a number. Change the number to set the color
Graphic effects
Graphic effects in Scratch can be used to change a sprite's appearance and its shape. They're fun to experiment with.

You'll never know where I'll appear next!
This block selects a random vertical position
set random -150 to 150
This block makes the ghost move slowly, hidden from view

13 Now update the main game loop to use the functions you have just created. Remember that the order is important, so make sure you put everything in the right place. Then run the code. Bubbles should burst when they hit the sub. Check the shell window to see the score.

```
score = 0
#MAIN GAME LOOP
while True:
    if randint(1, BUB_CHANCE) == 1:
        create_bubble()
    move_bubbles()
    clean_up_bubs()
    score += collision()
    print(score)
    window.update()
    sleep(0.01)
```

EXPERT TIPS
Python shortcut
The code "score += collision()" is a shortcut for writing "score = score + collision()". It adds the collision score to the total score, then updates the total score. Code like this is common, so a shortcut is useful. You can also do the same thing using the "--" symbol. For example, "score -= 10" is the same as "score = score - 10".



Each line of code is clearly labeled so you can't go wrong

This icon indicates that the project continues on the next page

Boxes give extra information: tips, definitions, and things to remember



EXPERT TIPS
When to save
This save icon appears on the project spreads. It reminds you when to save the work you've done, so that nothing is lost if the computer crashes. Always remember to save your work frequently.
Don't forget to save your work





- Lituz.com

Elektron kitoblar

**To'liq qismini Shu tugmani
bosish orqali sotib oling!**