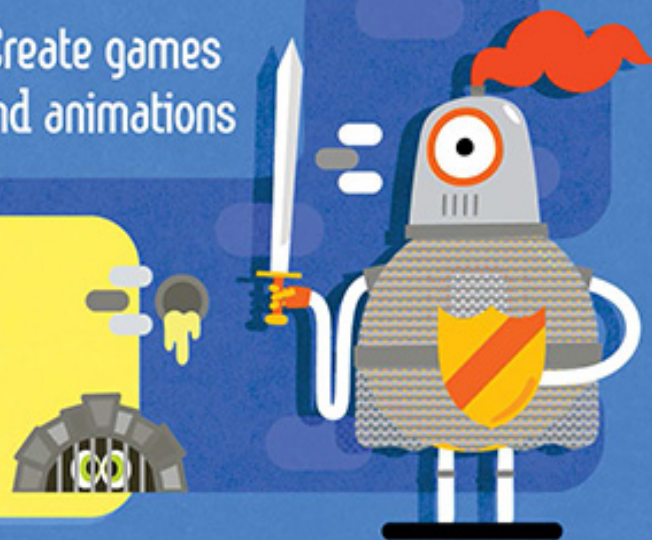


USBORNE



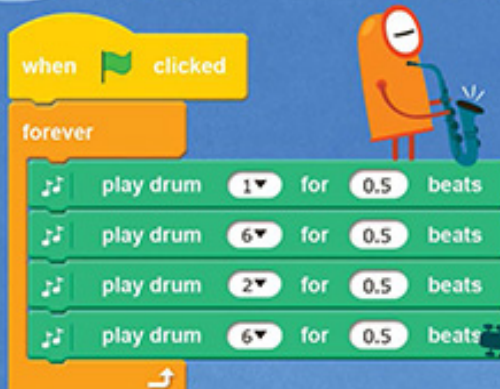
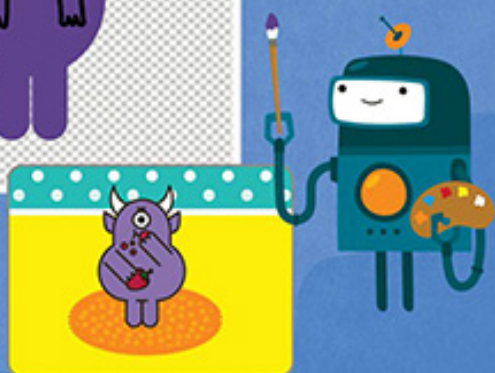
Create games
and animations



Coding for Beginners using SCRATCH



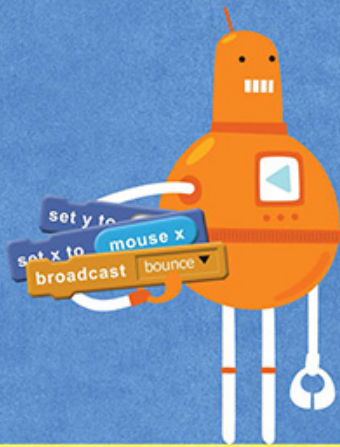
Paint your
own characters



Play sounds
and music

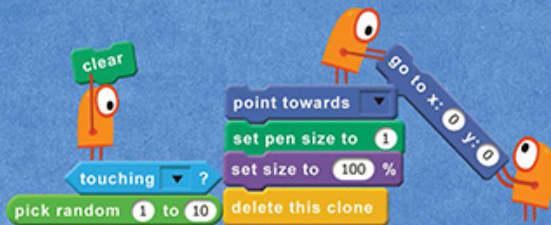


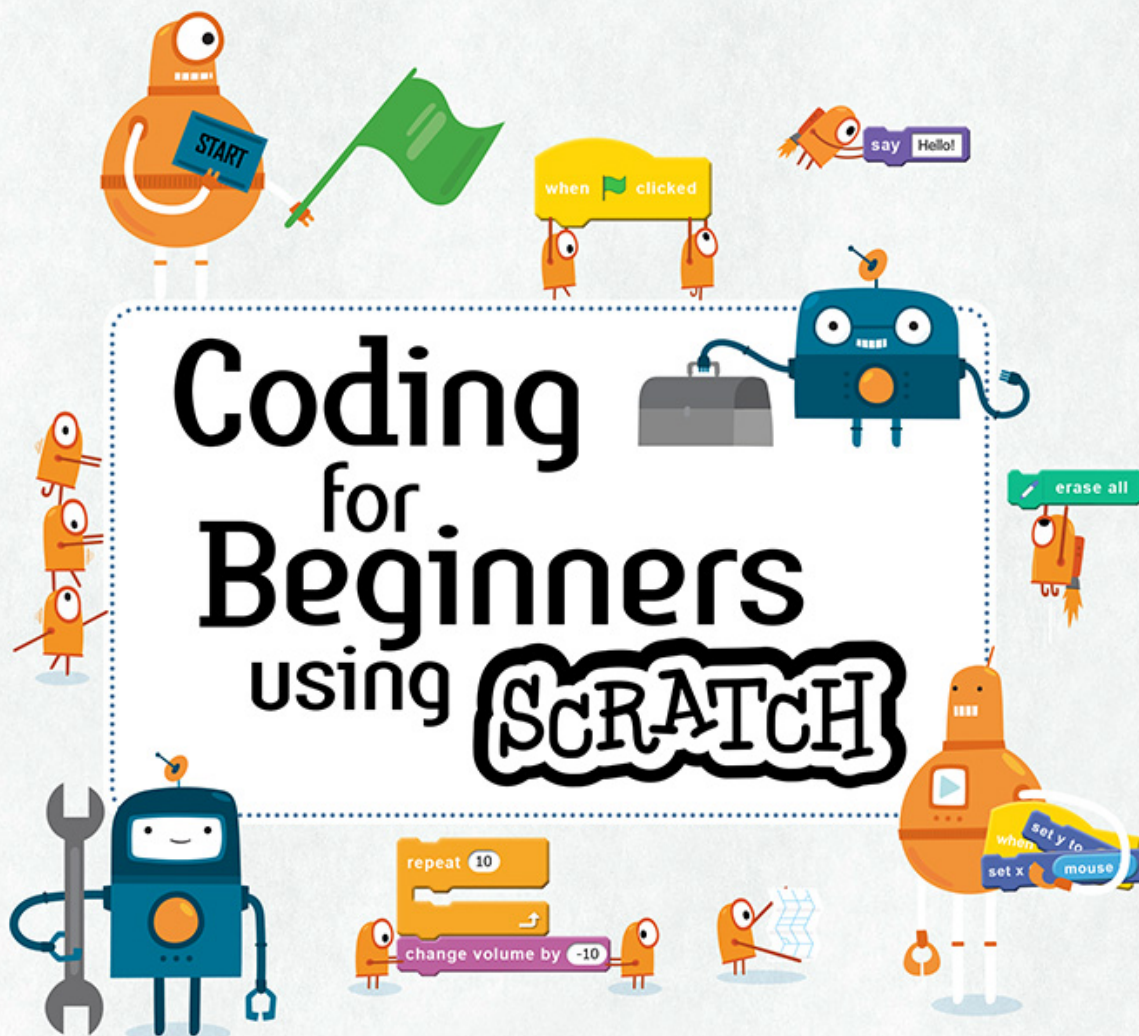
Revised and updated for Scratch 3



A complete and utter beginner's guide to coding, using the Scratch computer language.

- Step-by-step instructions for getting started, and for making games, animations and more
- Complete Scratch menu guides and glossary
 - Extra help and downloads online





Rosie Dickins,
Jonathan Melmoth & Louie Stowell

Coding consultants:
Berkbank Green & Jonathan Skuse
Additional coding advice: Ben Woodhall

Illustrated by Shaw Nielsen

Designed by Stephen Moncrieff,

Contents

What is coding? 4

Starting Scratch 6

First projects

Cat and mouse 8

Dancing sprites 12

Build a band 14

Boo! 18

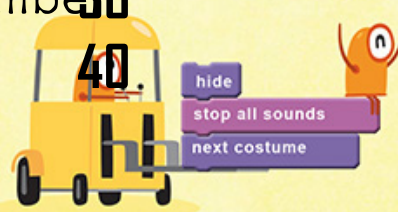
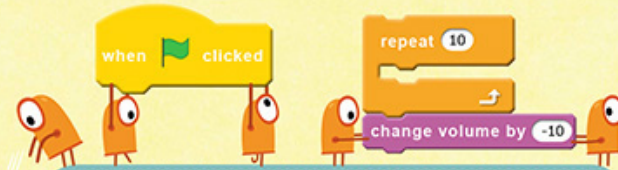
Drawing 22

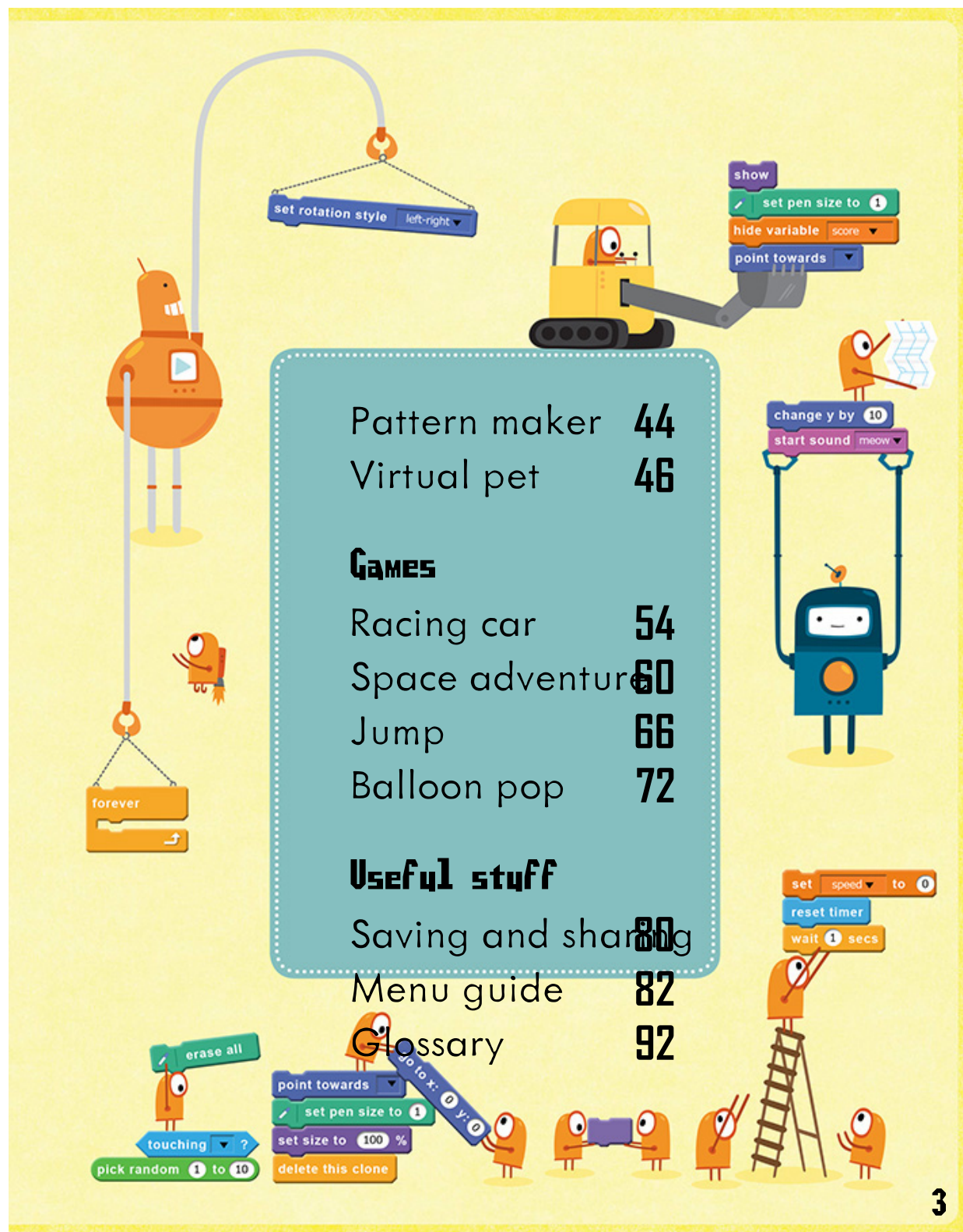
Once upon a time 26

Painting sprites 32

Guess the number 36

Bat and ball 40





What is coding?

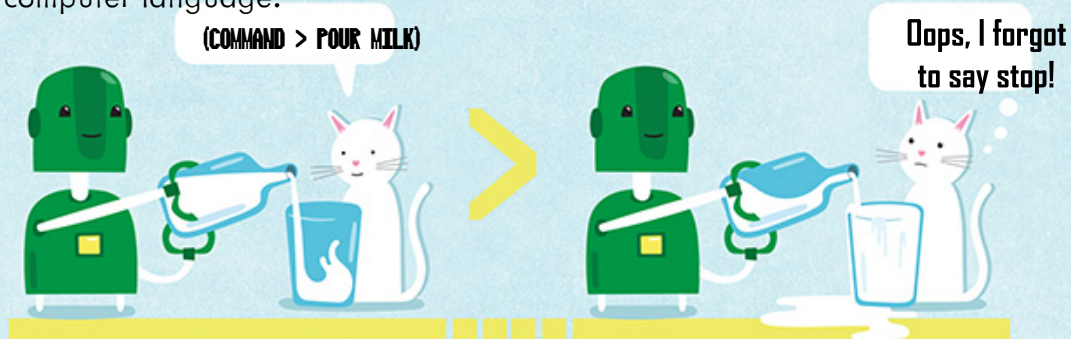
Coding means writing instructions for computers.
A finished set of instructions is known as a program.
If you learn to code, you can create programs of your own.

Being understood

For a program to work, it must be written a way that the computer understands. That means breaking down all the instructions into clear, simple steps, and putting them into computer language.

WARNING!

Computers follow instructions blindly – they can't think for themselves. So everything must be spelled out clearly, leaving nothing out.



Computer language

Computer language is like ordinary language, but with a limited word list and precise rules about how to set things out.

There are many different computer languages, designed for different kinds of coding. The first one most people learn is called *Scratch* – a language made especially for beginners.

Scratch is great for making games and animations – and for learning about coding in general.



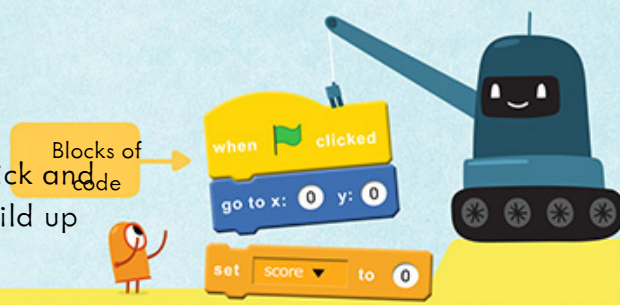
SCRATCH

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

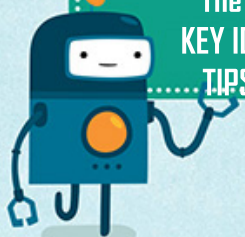


Why choose Scratch?

Scratch was designed to be quick and easy to use. It allows you to build up programs by slotting together ready-made blocks of code.



Look out for boxes like this one. The green boxes explain KEY IDEAS. Blue boxes have TIPS on using SCRATCH.



About this book

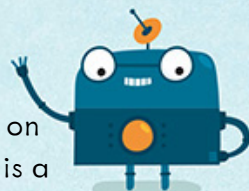
This book will show you how to make the most of Scratch by creating animations, stories and games – along with lots of tips for writing your own code. All the examples are broken down into short, easy-to-follow steps.

Getting started

The simplest way to use Scratch is on the Scratch website. All you need is a computer (one with a keyboard – not a tablet) and an internet connection.

Go to www.usborne.com/quicklinks and type in the name of this book, for a link to the Scratch website and full instructions, as well as other useful coding resources. You will also find a link to finished, working code for all the programs in this book.

Please follow the safety guidelines at the Quicklinks website when using the internet.



If you want to use Scratch offline (without being connected to the internet), you can download the language and save it onto your computer. Just follow the instructions

Starting Scratch

When you start up Scratch on your computer, you will see this screen.

On the Scratch website, click 'Create' (in the blue bar) to get here.



The **green flag** and **red button** are used as start and stop buttons.

SCRIPT AREA

You drag blocks...and stack them over here. A stack of blocks is known as a **script**.

You can rearrange blocks as much as you like. Clicking on a block allows you to drag and move that block as well as any attached below it.

Right-clicking allows you to delete blocks. You can also push blocks back to the menu to get rid of them.

STAGE

This is where you watch your code come to life.

SPRITE AREA

Each script is attached to a picture known as a **sprite** – which you can manage here.

These names are the **block menus** – see below for how they work.

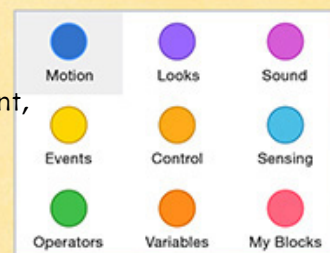
This button opens **Extensions** which gives you access to **music** and **pen blocks**.

You may have to close a tutorial box before you start. Click on the small 'x' to close them.

Block menus

Each **block menu** contains a variety of different, colour-coded blocks. For example...

- **Motion menu** blocks (blue) make sprites move.
- **Looks menu** blocks (purple) change how things look.
- **Control menu** blocks (orange) control the scripts themselves.



These are the nine **block menus**.

Click on a **block menu** name to bring up the blocks available, or turn to page 82 for a full list.

First steps

1 Try dragging these three blocks (from the **Motion** menu) into the **script area** to make the cat walk...

Then click on the **Sound** menu and add a **start sound** block.

2 Click on the script to **run** it. Click a few times and watch what happens.

The script glows as it runs, and the cat moves and meows. (If the cat goes too far, you can drag it back again.)

3 But the cat doesn't look as if it's walking. For that, its feet need to move...

Click on the **Looks** menu and add a **next costume** block.

This changes to another picture or 'costume' of the same sprite (in this case, the cat with its feet in a new position).

4 The cat's feet move – but only when you click on the script a few times.

To keep things going, you need to go to the **Control** menu for a **repeat** block. This block makes all the instructions inside it repeat, or **loop**, as many



Select 'mouse-pointer' from the drop-down menu.

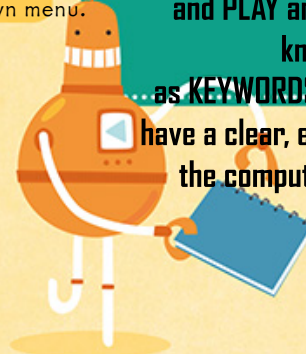


Select 'meow' from the drop-down menu.



KEYWORDS

Instruction words such as **MOVE** and **PLAY** are sometimes known as **KEYWORDS** because they have a clear, exact meaning in the computer language.



Congratulations, you've written your first piece of code!



You can type into the little white boxes to change the numbers.



LOOPS

LOOPS are used a lot in all kinds of code, because they make programs much shorter and quicker to write.

Turn the page to see how to turn this script into a simple cat-and-mouse game.

Cat and mouse

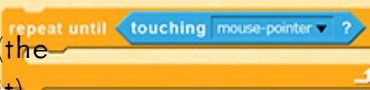
The aim of this game is to keep your mouse-pointer one step ahead of the cat. If the cat touches the mouse-pointer, it says, "Got you!" and the game is over.

1 This game requires a new kind of loop: **repeat until** (from the **Control** menu)...



...plus a diamond-ended block from the **Sensing** menu.

2 Snap them together (the loop will expand to fit).



Then

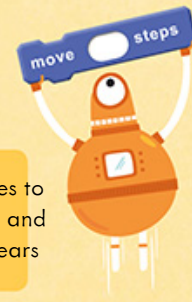
click on the black triangle and select 'mouse-pointer' from the drop-down menu.



This loop will make whatever is inside it repeat over and over, until the cat touches the mouse-pointer.

3 Go back to the script from the last page. Click on the first block in the loop, then drag that block, along with all the blocks attached below it, into your new loop.

Click on the white boxes to type the cat's message, and to set how long it appears on the screen.



Testing your script

4 Click on the script block and move your mouse-pointer around. The blocks should follow your mouse until it catches you. Try it a few times.

If the cat hits the edge of the stage, it flickers. But you can fix this by inserting a **bounce** (from the **Motion** menu) at the start of the loop.



CONDITIONALS

Instructions such as **IF** and **REPEAT UNTIL** tell the computer to react differently to different conditions (in this case, where the cat is).

So they are known as **CONDITIONAL**

instructions. If your code works, give yourself a pat on the back.

