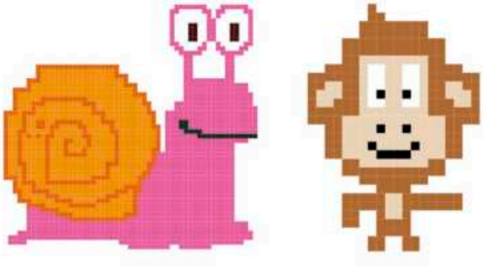
DK

coding PROJECTS IN SCRATCH™

A **step-by-step** visual guide to coding your own animations, games, simulations, and more!

coding PROJECTS IN SCRATCH™



coding PROJECTS IN SCRATCH™

JON WOODCOCK

DK Penguin Random House

DK UK

Senior editor Ben Morgan
Senior art editor Jacqui Swan
US editors Jill Hamilton, Margaret Parrish
Jacket design development manager Sophia MTT
Jacket editor Claire Gell
Producer, pre-production Gillian Reid
Producer Mary Slater
Managing editor Lisa Gillespie
Managing art editor Owen Peyton Jones
Publisher Andrew Macintyre
Associate publishing director Liz Wheeler
Art director Karen Self
Design director Phil Ormerod
Publishing director Jonathan Metcalf

DK DELHI

Project editor Suefa Lee
Project art editor Parul Gambhir
Editor Sonia Yooshing
Art editor Sanjay Chauhan
Assistant art editor Sonakshi Singh
Jacket designer Suhita Dharamjit
Managing jackets editor Saloni Singh
DTP designer Jaypal Chauhan
Senior managing editor Rohan Sinha
Managing art editor Sudakshina Basu
Pre-production manager Balwant Singh

First American Edition, 2016
Published in the United States by DK Publishing
345 Hudson Street, New York, New York 10014

16 17 18 19 20 10 9 8 7 6 5 4 3 2 1
001—289247—July/2016

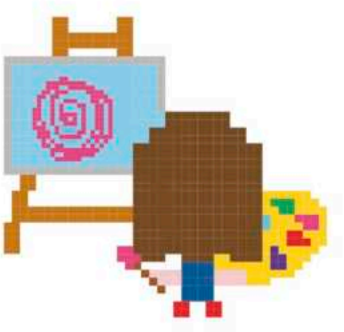
Published in Great Britain by Dorling Kindersley Limited.

A catalog record for this book is available from the Library of Congress.
ISBN: 978-1-4654-5142-2

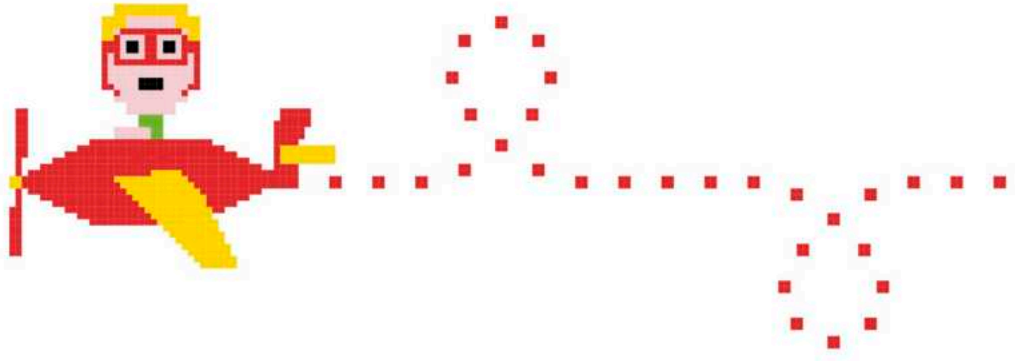
Printed in China

A WORLD OF IDEAS:
SEE ALL THERE IS TO KNOW

www.dk.com

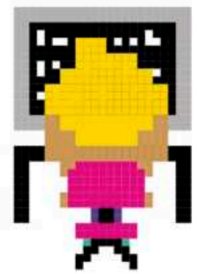


DR. JON WOODCOCK MA (OXON) has a degree in physics from the University of Oxford and a Ph.D. in computational astrophysics from the University of London. He started coding at the age of eight and has programmed all kinds of computers, from single-chip microcontrollers to world-class supercomputers. His 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 many science and technology books, and is coauthor of DK's *Help Your Kids with Computer Coding* and author of DK's *Coding Games in Scratch*, and DK's series of coding workbooks.

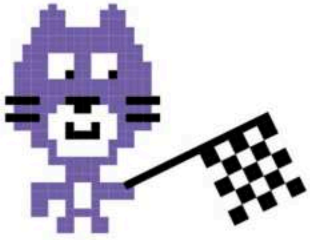


Contents

1 WHAT IS CODING?

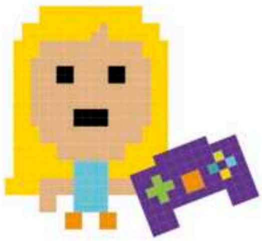


2 GETTING STARTED



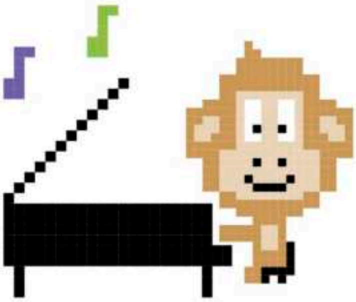
3 ART

4 GAMES



5 SIMULATIONS

6 MUSIC AND SOUND



7 MINDBENDERS

8 WHAT NEXT?

Find out more at:
www.dk.com/computercoding

Foreword

In recent years, interest in coding has exploded. All over the world, schools are adding coding to their curriculums, code clubs are being launched to teach beginners, and adults are returning to college to learn coding skills now considered vital in the workplace. And in homes everywhere, millions of people are learning how to code just for the fun of it.

Fortunately, there's never been a better time to learn how to code. In the past, programmers had to type out every line of code by hand, using obscure commands and mathematical symbols. A single period out of place could ruin everything. Today, you can build amazingly powerful programs in minutes by using drag-and-drop coding languages like Scratch™, which is used in this book.

As learning to code has become easier, more people have discovered the creative potential of computers, and that's where this book comes in. *Coding Projects in Scratch* is all about using code for creative purposes—to make art, music, animation, and special effects. With a little bit of imagination you can produce dazzling results, from glittering fireworks displays to kaleidoscope-like masterpieces that swirl and beat in time to music.

If you’re completely new to coding, don’t worry—the first two chapters will walk you through the basics and teach you everything you need to know to use Scratch. The later chapters then build on your skills, showing you how to create interactive artworks, lifelike simulations, mind-bending optical illusions, and some great games.

Learning something new can sometimes feel like hard work, but I believe you learn faster when you’re having fun. This book is based on that idea, so we’ve tried to make it as much fun as possible. We hope you enjoy building the projects in this book as much as we enjoyed making them.

1

What is coding?

Creative computers

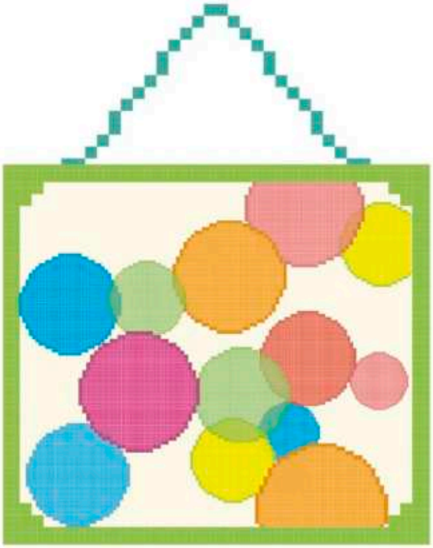
Computers are everywhere and are used in all sorts of creative ways. But to really join in the fun, you need to take control of your computer and learn how to program it. Programming puts a world of possibilities at your fingertips.

Think like a computer

Programming, or coding, simply means telling a computer what to do. To write a program you need to think like a computer, which means breaking down a task into a series of simple steps. Here's how it works.

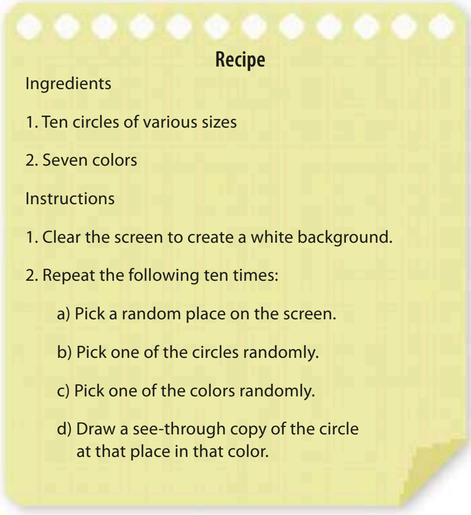
▷ **A simple recipe**

Imagine you want a friend to bake a cake, but your friend has no idea how to cook. You can't simply give an instruction like "make a cake"—your friend won't know where to start. Instead, you need to write a recipe, with simple steps like "break an egg," "add the sugar," and so on. Programming a computer is a bit like writing a recipe.



◁ **Step by step**

Now imagine you want to program a computer to create a painting like the one shown here, with colored circles overlapping each other at random. You have to turn the job of painting the picture into a kind of recipe, with steps the computer can follow. It might look something like this:



Recipe

Ingredients

1. Ten circles of various sizes

2. Seven colors

Instructions

1. Clear the screen to create a white background.

2. Repeat the following ten times:

    a) Pick a random place on the screen.

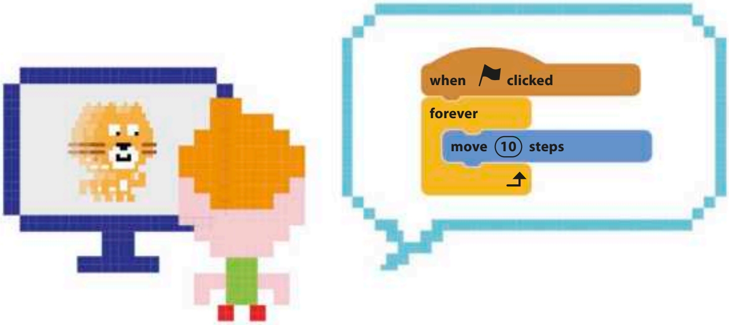
    b) Pick one of the circles randomly.

    c) Pick one of the colors randomly.

    d) Draw a see-through copy of the circle at that place in that color.

▷ **Computer language**

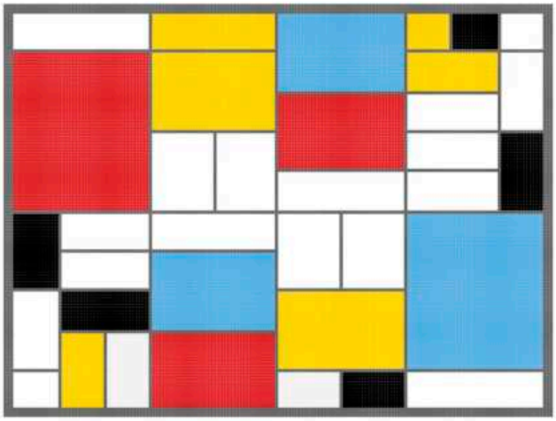
Although you can understand the recipe for a painting or a cake, a computer can't. You need to translate the instructions into a special language that the computer can understand—a programming language. The one used in this book is called Scratch.



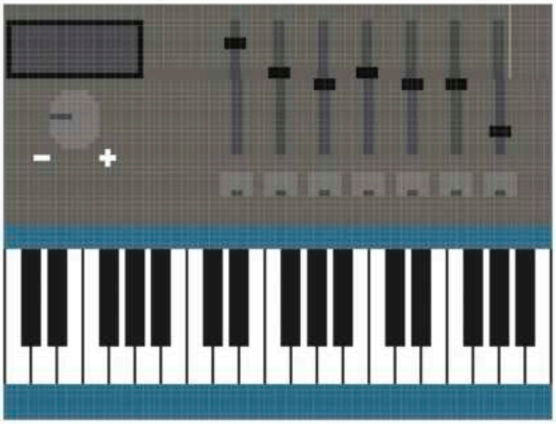
```
when ⚑ clicked
forever
    move (10) steps
```

Worlds of imagination

There isn't a single creative field in the world that hasn't been touched by computers. In this book, you'll get to make lots of great projects that will fire your imagination and make you think and code creatively.



Computers can be programmed to create original works of art.



Sound programs can mix musical and other sound effects in any combination.

Building games programs is just as much fun as playing them, especially when you make all the rules.

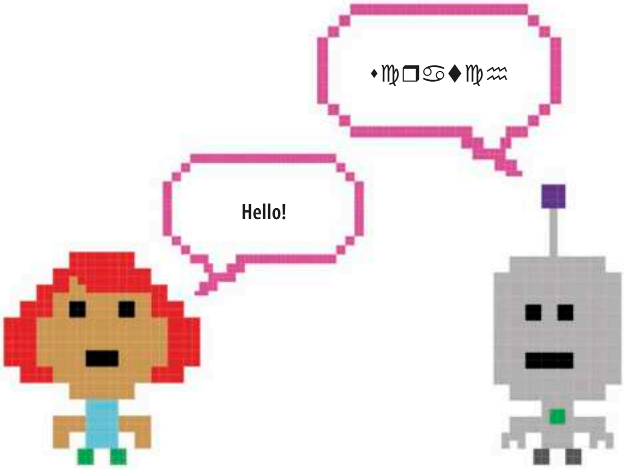
Special effects and dramatic scenery in movies are often created in graphics programs.

Programming languages

To tell a computer what to do, you need to speak the right kind of language: a programming language. There are lots to choose from, ranging from easy ones for beginners, like the one in this book, to complex languages that take years to master. A set of instructions written in any programming language is called a program.

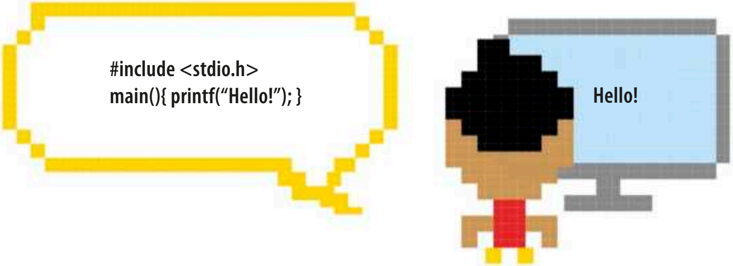
Popular languages

There are more than 500 different programming languages, but most programs are written in just a handful of these. The most popular languages use English words, but lines of code look very different from English sentences. Here's how to get a computer to say "Hello!" on screen in just a few of today's languages.



▷ C

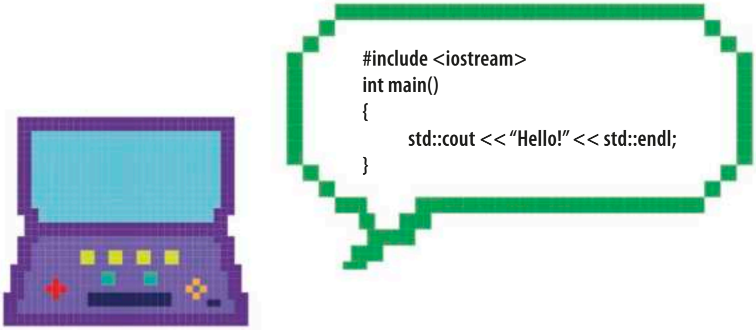
The C programming language is often used for code that runs directly on a computer's hardware, such as the Windows operating system. C is good for building software that needs to run fast, and has been used to program space probes.



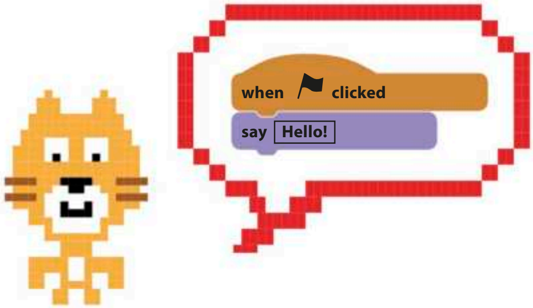
```
#include <stdio.h>
main(){ printf("Hello!"); }
```

▷ C++

This complicated language is used to build large, commercial programs such as word processors, web browsers, and operating systems. C++ is based on C, but with extra features that make it better for big projects.



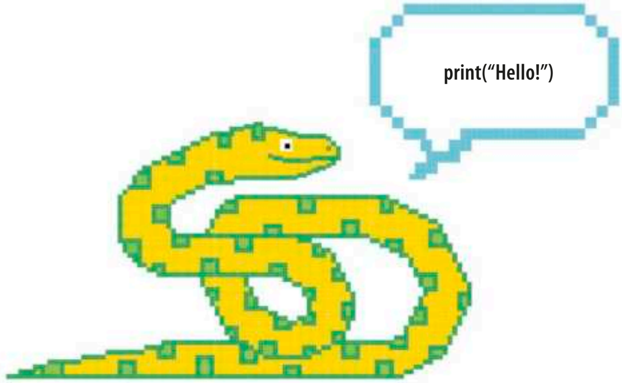
```
#include <iostream>
int main()
{
        std::cout << "Hello!" << std::endl;
}
```



```
when ⚑ clicked
say Hello!
```

△ **Scratch**

Beginners often start with simple programming languages such as Scratch. Instead of typing out code, you build scripts with ready-made blocks of code.



```
print("Hello!")
```

△ **Python**

Python is a very popular, all-purpose language. The lines of code are shorter and simpler than in other languages, making it easier to learn. Python is a great language to learn after Scratch.

```
alert('Hello');
```

△ **JavaScript**

Programmers use JavaScript to create interactive features that run on websites, such as advertisements and games.

```
class HelloApp {
    public static void main(String[] args) {
        System.out.println("Hello!");
    }
}
```

△ **Java**

Java code is designed to work on all types of devices, from cell phones and laptops to games consoles and supercomputers. Minecraft is written in Java.

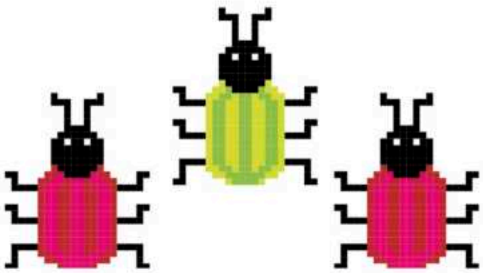
LINGO

Code words

Algorithm A set of instructions that are followed to perform a particular task. Computer programs are based on algorithms.

Bug A mistake in a program. They are called bugs because the first computers had problems when insects got stuck in their circuits.

Code Computer instructions written in a programming language are often called code. Coding is programming.

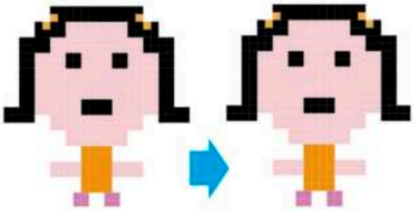


How Scratch works

This book shows you how to build some really cool projects using the Scratch programming language. Programs are made by dragging together ready-made blocks of instruction code to control colorful characters called sprites.

Sprites

Sprites are the objects shown on the screen. Scratch comes with a huge selection of sprites—such as elephants, bananas, and balloons—but you can also draw your own. Sprites can perform all sorts of actions, like moving, changing color, and spinning around.



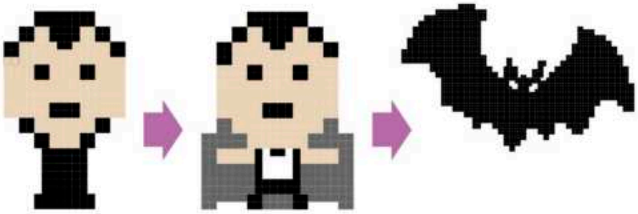
Sprites can move around.

Sprites can play sounds and music.

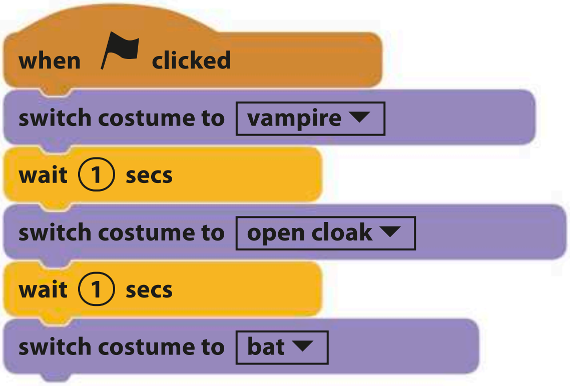
Sprites can deliver messages on the screen.

Blocks and scripts

Scratch's multicolored instruction blocks tell sprites what to do. Each sprite gets its instructions from stacks of Scratch blocks called scripts. Each instruction block is performed in turn from the top to the bottom. Here's a simple script for this vampire sprite.



▽ **Creating scripts**
The blocks that make a script are dragged together using a computer mouse. They lock together like pieces of a jigsaw puzzle. Blocks come in color-coded families to help you find the correct block easily. For example, all the purple blocks change a sprite's appearance.



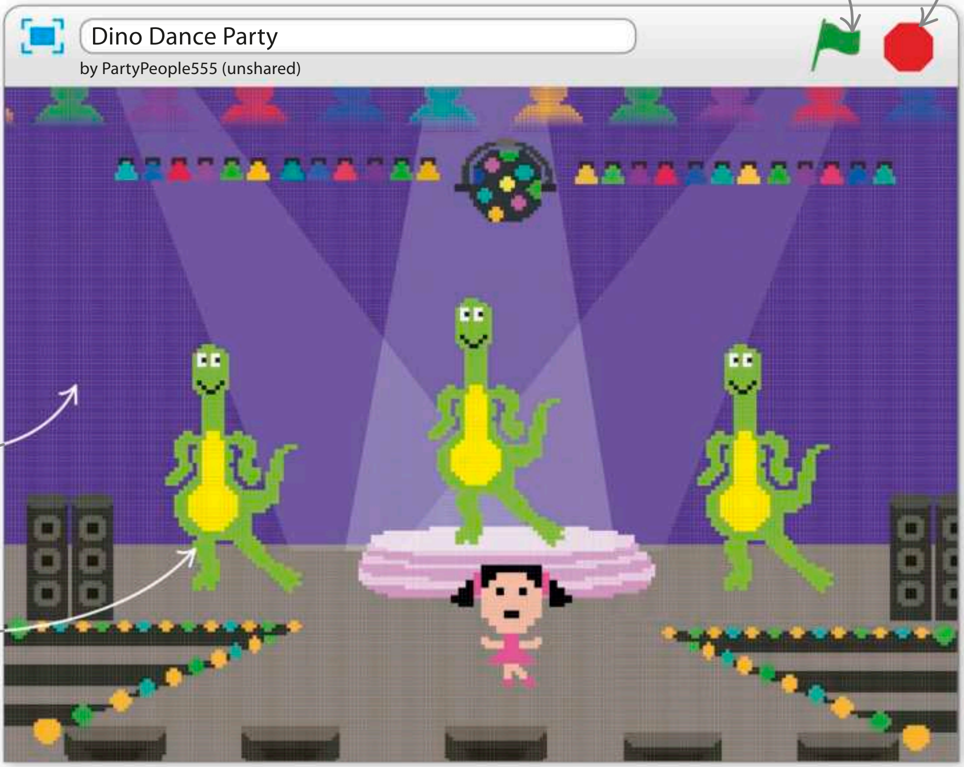
```
when ⚑ clicked
switch costume to vampire ▼
wait 1 secs
switch costume to open cloak ▼
wait 1 secs
switch costume to bat ▼
```

A typical Scratch project

A Scratch project is made up of sprites, scripts, and sounds, which work collectively to create action on the screen. The area where you see the action is called the stage. You can add a background picture called a backdrop to the stage.

▷ Green for go!

Starting, or "running," a program brings to life the scripts you've built. In Scratch, clicking the green flag runs all the scripts in the project. The red button stops the scripts so you can continue working on your code.



The red button stops a program.

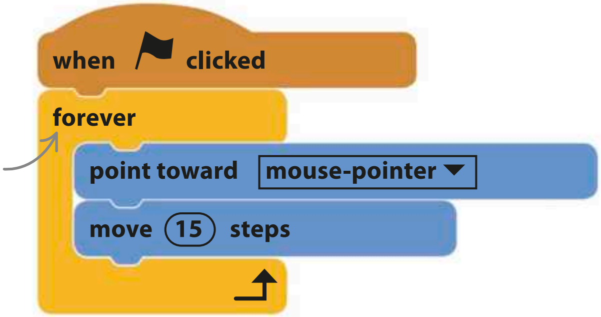
The green flag starts the program.

The stage and lights are part of the backdrop (background picture).

The dancing dinosaurs and ballerina are sprites controlled by their own scripts.

▽ Scripts work together

A project usually has several sprites, each with one or more scripts. Each script creates just a part of the action. This script makes a sprite chase the mouse-pointer around the stage.



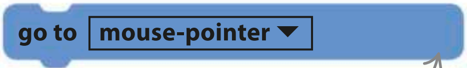
```
when 🏴 clicked
forever
    point toward mouse-pointer ▾
    move (15) steps
```

The "forever" block makes the blocks inside repeat.

EXPERT TIPS

Reading Scratch

Scratch is designed to be easily understood. The action performed by each block is written on it, so you can usually figure out what a script does just by reading through it.



```
go to mouse-pointer ▾
```

Can you guess what this block makes sprites do?