Why choose this book?

£10.99 UK / $14.99 US / $16.95 CN

- We don't just tell you how to do it, we show you how.
- It's fully illustrated.
- Put off by the jargon? Don't worry, we show you how.
- It's easy to follow.
- Do we need to say any more?
- And it's fantastic value.
- This book's printed in color to make it in full color.
- It's in full color.
- Keep things straight forward.
- Wherever you see one of these icons you know there's a handy tip to spice up your learning, flag something to remember or ward you away from potential dangers.

This primer introduces you to Scratch Programming in easy steps.

By the time you've finished, you'll be impressing your friends and family with your own computer games!

We use the “broadcast and wait” block to stop the player from being able to fire a second missile while the first one is still on the screen.

Adding a script for when the player presses the space bar to fire, we'll add a script to the player's sprite that broadcasts “fire” when the player presses the space bar, and we'll tell our missile sprite to move when it receives that broadcast:

Add a script for when the alien sprite directly looks at the player.

Add a script to your Scratch program that makes the player invisible for an instant when it hits the alien.

Add this script to your Scratch program:

Click the ship in the Sprite List. Add the set rotation style left-right script to the sprite.

You can have several scripts on the same sprite. Mine is 50%.

Click the alien in the Sprite List. Add the change direction script to the sprite. Make sure that the alien sprite to move around the stage changes direction when it reaches the stage's edge. It is the program to move the sprite to a random position on the right of the Stage by changing direction, making eight different colors, drawn with the pen.

This is a programming language that is widely used on the Raspberry Pi and in schools and colleges.

Let these icons make it even easier.

Moving and shooting aliens

The alien uses the pixelate effect to materialize in a random position on the right. It makes it easier for the player to avoid hitting them, but still miss the missile to hit them too. Smaller aliens make it even easier.
Learn to program with Scratch

Scratch makes it easy to create your own games, animations, music, art or applications. It’s the perfect way to learn programming because it takes away a lot of the complexity. That means you can focus on having great ideas and bringing them to life. With this book as your companion, you’ll learn how to:

- Design, build and share your own programs
- Create addictive arcade games, quizzes and word games
- Make computer-generated art
- Play your favourite music and compose your own tunes
- Use variables, lists, loops, broadcasts and operators to create sophisticated software
- Avoid common programming pitfalls and bugs
- Interact with webcam video and the sensors on a PicoBoard

This book is an in-depth guide to both the new web-based Scratch 2.0, and Scratch 1.4, as used on the Raspberry Pi. With no prior knowledge needed, this book takes you from scratch to Scratcher!
Foreword by Mitchel Resnick 7

1 Introducing Scratch 9

- What is Scratch? 10
- Which version of Scratch? 11
- Starting on the Raspberry Pi 12
- Creating a Scratch account 13
- Using the Scratch screen 14
- Exploring the blocks 16
- Changing the backdrop 18
- Creating your first program 20
- Saving your project 23
- Opening projects 25
- Opening shared projects 26

2 Drawing with Scratch 27

- Understanding coordinates 28
- Changing a sprite’s position 30
- Using the pen 31
- Drawing a house in Scratch 33
- Using directions to move 35
- Keeping sprites upright 36
- Drawing using directions 38
- Making shapes using Repeat 39
- Putting loops inside loops 40
- Creating Rainbow Painter 41

3 Spiral Rider 43

- Introducing Spiral Rider 44
- Using the green flag 45
- Creating variables 46
- Drawing a spiral 48
- Changing the background 51
- Adding sprites 52
- Animating the crab 54
- Enabling keyboard control 56
- Making the fish move 58
- Adding Game Over messages 61
### Super Dodgeball

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing Super Dodgeball</td>
<td>64</td>
</tr>
<tr>
<td>Setting up the variables</td>
<td>65</td>
</tr>
<tr>
<td>Preparing for the game start</td>
<td>67</td>
</tr>
<tr>
<td>Using coordinates to move</td>
<td>68</td>
</tr>
<tr>
<td>Adding more images</td>
<td>70</td>
</tr>
<tr>
<td>Renaming sprites</td>
<td>71</td>
</tr>
<tr>
<td>Making random numbers</td>
<td>72</td>
</tr>
<tr>
<td>Moving the ball</td>
<td>73</td>
</tr>
<tr>
<td>Copying and deleting sprites</td>
<td>75</td>
</tr>
<tr>
<td>Adding the energy meter</td>
<td>78</td>
</tr>
<tr>
<td>Painting in Scratch</td>
<td>80</td>
</tr>
<tr>
<td>Using Paint tools</td>
<td>82</td>
</tr>
<tr>
<td>Using colors</td>
<td>84</td>
</tr>
<tr>
<td>Using vectors in Scratch 2.0</td>
<td>85</td>
</tr>
<tr>
<td>Making the ice cream appear</td>
<td>86</td>
</tr>
<tr>
<td>Enabling the player to score</td>
<td>88</td>
</tr>
<tr>
<td>Tweaking the gameplay</td>
<td>90</td>
</tr>
</tbody>
</table>

### Cosmic Chorus

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing Cosmic Chorus</td>
<td>92</td>
</tr>
<tr>
<td>Adding sprites and costumes</td>
<td>93</td>
</tr>
<tr>
<td>Preparing Cosmic Chorus</td>
<td>96</td>
</tr>
<tr>
<td>Adding sounds to a sprite</td>
<td>97</td>
</tr>
<tr>
<td>Sounds in Scratch 1.4</td>
<td>99</td>
</tr>
<tr>
<td>Recording in Scratch 1.4</td>
<td>100</td>
</tr>
<tr>
<td>Playing sounds</td>
<td>101</td>
</tr>
<tr>
<td>Making Gobo sing</td>
<td>102</td>
</tr>
<tr>
<td>Using broadcasts</td>
<td>103</td>
</tr>
<tr>
<td>Conducting the Gobo</td>
<td>104</td>
</tr>
<tr>
<td>Adding more Gobos</td>
<td>105</td>
</tr>
<tr>
<td>Playing music in Scratch</td>
<td>107</td>
</tr>
<tr>
<td>Adding the singing soloist</td>
<td>108</td>
</tr>
<tr>
<td>Adding a title screen</td>
<td>110</td>
</tr>
<tr>
<td>Enabling the title screen</td>
<td>111</td>
</tr>
<tr>
<td>Making your own tunes</td>
<td>112</td>
</tr>
<tr>
<td>London Bridge</td>
<td>114</td>
</tr>
</tbody>
</table>
6 Quiz Break
Introducing Quiz Break 116
Preparing for Quiz Break 117
Drawing the timer 119
Moving the clock hand 121
Adding a rising sound effect 124
Asking questions 125
Joining text to greet players 126
Preparing the quiz 127
Making the questions 128
Checking the answers 130
Reporting the scores 132
Adding the victory dance 133

7 Hangman
Introducing Hangman 136
Importing the images 137
Making the gallows 138
Making the alphabet sprite 140
Creating the variables 141
Setting up the gallows 142
Losing the game 143
Adding graphic effects 144
Creating the main script 146
Creating lists 149
Using lists 150
Setting up the variables 152
Showing the game board 155
Testing the game board 158
Asking for the player’s guess 159
Checking the letter 161
Game over 164

8 Space Swarm
Introducing Space Swarm 166
Preparing the images 167
Adding sound effects 169
Creating the variables 170
Programming the hero 171
Enabling the player to fire 174
Moving and shooting aliens 175
Finishing touches 177
Adding the high score 178
How to use this book

Welcome to the wonderful world of Scratch!

This book will show you some of the great projects you can build with Scratch, and will give you the skills and knowledge you need to start designing your own projects.

The book has been written so that each chapter teaches you something new, and builds on the previous chapters. For that reason, the best way to use this book is to work through the chapters in the right order.

Feel free to experiment as you go, though: there are lots of things you can do to customize and improve the examples in this book, and that’s a big part of what Scratch is all about.

If you need advice on something that hasn’t been covered yet, or you want to revise something you read earlier, consult the index for more information.

Here’s what’s in store:

- In Chapter 1, you’ll learn the basics of Scratch. Sprites are objects you can use in your games, and you’ll meet the cat sprite here. In your first program, it goes for a quick walk.

- In Chapter 2, you’ll learn how to draw with Scratch, how to move sprites around the Stage, and how to make bits of your program repeat. The final project is art program Rainbow Painter, which leaves a colorful line behind your mouse.

- In Chapter 3, Spiral Rider is your first game. You’ll learn how to enable keyboard control of a sprite, and how to detect when the player’s character touches something else, and you’ll discover how to remember information in a program.

- In Chapter 4, the Dodgeball game shows you how to build a game with lots of sprites moving at once. You’ll learn how to design your own sprites, how to clone them in Scratch 2.0, how to make random numbers, and how to keep score.

- In Chapter 5, you get to conduct the Cosmic Chorus. You’ll see how to use recorded sounds and musical notes, how to get one sprite to control another, and how to make a title screen.
In Chapter 6, Quiz Break tests players’ maths skills against the clock. You’ll learn how to use the timer, how to use the operators for maths, and how to join text together.

In Chapter 7, the game Hangman teaches you how to use lists in Scratch, how to ask the player for information, how to structure complex programs, and how to write on the Stage.

In Chapter 8, you’ll create an arcade game and learn how to make one sprite fire on another, and how to make special effects to include in your game.

In Chapter 9, you’ll learn how to use Scratch with your webcam, microphone and the PicoBoard add-on which combines several sensors in one Scratch-friendly device. Flap your hands at the screen to wave away the bats and use your Picoboard to control a witch’s night flight. You’ll also see how to control a light connected to the Raspberry Pi from Scratch.

Chapter 10 contains seven short programs you can experiment with and customize, including the games Shop Cat, Maze Mania, Photo Safari and Breakthrough.

Chapter 11 gives you advice on creating your own programs, and resources you can use to help you get started.

**Downloading the examples**

The best way to use this book is to work your way through the chapters in order, creating the examples as you go. You’ll learn more about programming, and you’ll have a chance to customize the examples as you go.

If you can’t get the examples to work, or if you want to take a shortcut, you can download them. The examples for Scratch 1.4 can be downloaded at the author’s website at [www.sean.co.uk](http://www.sean.co.uk) and the publisher’s website at [www.easysteps.com](http://www.easysteps.com)

To find the examples for Scratch 2.0, visit the author’s profile on the Scratch website. You can find it at [http://scratch.mit.edu/users/seanmcmanus/](http://scratch.mit.edu/users/seanmcmanus/)
Is it important for all children to learn how to write? After all, very few children grow up to become journalists, novelists, or professional writers. So why should everyone learn to write?

Of course, such questions seem silly. People use writing in all parts of their lives: to send birthday messages to friends, to jot down shopping lists, to record personal feelings in diaries. The act of writing also engages people in new ways of thinking. As people write, they learn to organize, refine, and reflect on their ideas. It’s clear that there are powerful reasons for everyone to learn to write.

I see coding (computer programming) as an extension of writing. The ability to code allows you to “write” new types of things – interactive stories, games, animations, and simulations. And, as with traditional writing, there are powerful reasons for everyone to learn to code.

The recent surge of interest in learning to code, reflected in sites like codecademy.com and code.org, has focused especially on job and career opportunities. It is easy to understand why: the number of jobs for programmers and computer scientists is growing rapidly, with demand far outpacing supply.

But I see much deeper and broader reasons for learning to code. In the process of learning to code, people learn many other things. They are not just learning to code, they are coding to learn. In addition to learning mathematical and computational ideas (such as variables and conditionals), they are also learning strategies for solving problems, designing projects, and communicating ideas. These skills are useful not just for computer scientists but for everyone, regardless of age, background, interests, or occupation.

Six years ago, my research group at the MIT Media Lab launched the Scratch programming language and online community in an effort to make coding accessible and appealing to everyone.
We’ve been amazed with the diversity and creativity of the projects. Take a look at the Scratch website and you’ll find animated stories, virtual tours, science simulations, public-service announcements, multimedia art projects, dress-up games, paint editors, and even interactive tutorials and newsletters.

We find that active members of the Scratch community start to think of themselves differently. They begin to see themselves as creators and designers, as people who can make things with digital media, not just browse, chat, and play games. While many people can read digital media, Scratchers can write digital media.

Scratch community members also begin to see the world in new ways. As one 11-year-old Scratcher wrote on a public blog: “I love Scratch. Wait, let me rephrase that – Scratch is my life. I have made many projects. Now I have what I call a ‘Programmer’s mind.’ That is where I think about how anything is programmed. This has gone from toasters, car electrical systems, and soooo much more.”

It has been exciting to watch what young people are creating and learning with Scratch. But this is just the beginning. The new version of the Scratch programming language and online community moves Scratch into the cloud, enabling people to program, save, share, and remix Scratch projects directly in a web browser. The new version also adds many new features to enhance opportunities for creativity and collaboration.

But we are aware that new features and capabilities are not enough. The biggest challenges for the future are not technological but cultural and educational. Ultimately, what is needed is a shift in mindsets, so that people begin to see coding not only as a pathway to good jobs, but as a new form of expression and a new context for learning.

Mitchel Resnick

This is an edited version of an article that was originally published at Edsurge (www.edsurge.com).
In this chapter, you’ll get started with Scratch, including learning about the different versions, meeting some of the blocks used to give instructions, and creating your first program. You also learn how to save your work, and load example projects by others.

10 What is Scratch?
11 Which version of Scratch?
12 Starting on the Raspberry Pi
13 Creating a Scratch account
14 Using the Scratch screen
16 Exploring the blocks
18 Changing the backdrop
20 Creating your first program
23 Saving your project
25 Opening projects
26 Opening shared projects
What is Scratch?

Programming is the art of writing instructions to tell a computer what to do. A set of instructions is called a program. The instructions are written in what’s known as a programming language, and there are thousands to choose from.

Scratch is a programming language that is perfect for making games, animations, interactive stories and other visually rich programs. It provides a great introduction to programming for people of all ages. It’s widely used in schools and colleges, but Harvard University has also used it in higher education at its Summer School. I’ve led workshops for adults where Scratch provided a friendly introduction to the kind of creative problem solving that programmers do all the time.

Scratch is easier to use than most other programming languages for a number of reasons:

- You don’t have to remember or type any commands: they’re all on screen, so you can just drag and drop them.
- Commands fit together like jigsaw pieces, so there are strong visual hints about how you can combine them.
- Error messages are rare. Because Scratch commands lock together, programs always make some kind of sense. It is possible to still write programs with logical errors in, if they don’t do what you expected, but Scratch guides you to write things that work, rather than nagging you when they don’t.
- The commands are color-coded and categorized, so you can easily find a command when you need it.
- The commands in Scratch simplify common activities in games, such as testing whether a missile has hit an alien (collision detection), or rotating a character on screen.

In short, Scratch is designed for your success. It enables you to quickly see results from your work, and even includes graphics and sounds you can use so you can get started right now. Many other programming languages require you to learn text commands, and strict rules about how you can use them. Scratch doesn’t. That means you can focus your energy instead on the fun stuff: dreaming up ideas for new programs, working out how to build them, designing them, and sharing them with friends.
Which version of Scratch?

There are two versions of Scratch that are in popular use. In this book, I’ll tell you about both.

**Scratch 2.0**

Scratch 2.0 is the latest version of Scratch, which was introduced in May 2013. I recommend you use Scratch 2.0 if you can. This version makes it easier for people to share their projects, take a look at each other’s projects, and adapt them (or “remix” them).

You don’t need to install any software to use Scratch 2.0: it runs inside your browser, using an Internet connection. Scratch 2.0 needs a computer that can run the Adobe Flash Player, though, so it doesn’t work on some mobile devices (including the iPhone and iPad) and some lower-powered devices (including the Raspberry Pi). Most other computers run Scratch 2.0 fine.

There is also a version of Scratch 2.0 that you can download from the Scratch website and install on your computer, so you can use it without an Internet connection. There are no official plans to bring Scratch 2.0 to the Raspberry Pi, though (*correct at the time of printing*).

**Scratch 1.4**

Scratch 1.4 is the previous version of Scratch, and it is software you install on your computer. You can download it for Windows, Mac and Linux at [http://scratch.mit.edu/scratch_1.4/](http://scratch.mit.edu/scratch_1.4/)

This version of Scratch found a whole new audience with the meteoric rise of the Raspberry Pi, the stripped-down Linux-based computer for hobbyists and education (see photo).

---

**Beware**

If you’re using the Scratch 2.0 website, I recommend using the Google Chrome browser with it. I experienced some bugs using Internet Explorer, which disappeared when I started using Chrome. You can download Chrome for free at [www.google.com/chrome](http://www.google.com/chrome).

**Hot tip**

Members of the Scratch team have said that they hope to make an iPad version of Scratch, but that it might be more limited than Scratch 2.0.

**Hot tip**

If you have a weak Internet connection, or none at all, you might prefer to use the downloadable version of Scratch to the browser-based version.
Introducing Scratch

Starting on the Raspberry Pi

Scratch is included with the recommended version of the Linux operating system, called Raspbian. I’ll assume you’ve got your Raspberry Pi connected up and working already:

1. Switch on your Raspberry Pi and log in. The default login is `pi`, and the default password is `raspberry`.

2. When you see the prompt `pi@raspberrypi ~ $`, enter `startx` and press the Enter or Return key.

3. The LXDE desktop appears (see picture). The large Raspberry is just a background image, so you might see something different. You might also see different icons.

4. Double-click the Scratch icon, which has a picture of a cat on it. In the screenshot below, it’s in the top-left corner of the screen.

5. You can also start Scratch using the Programs Menu, as also shown below: click the button in the bottom-left to open the menu, click the Programming folder, and finally, click Scratch.

The commands on the Raspberry Pi are case-sensitive, so it won’t work if you type in `Startx` (with a capital S), for example.

**Hot tip**

If you don’t have Scratch installed on your Raspberry Pi, you can install it. After logging in, enter the command:

```bash
sudo apt-get update
```

```bash
&& sudo apt-get install scratch
```

That command will also update Scratch to the latest version, if Scratch is already installed.
Creating a Scratch account

Before you begin to program with Scratch 2.0, I recommend you create an account for the Scratch website. Here’s how:

1. Open a web browser, such as Google Chrome

2. Enter the website address http://scratch.mit.edu/ in your address bar, usually at the top of the screen

3. Click Join Scratch in the top-right

4. Make up a username. Scratch is used by young people (among others), so the site advises members to protect their privacy by not using real names. You can’t change your username later, so choose wisely

5. Pick a password and enter it twice. The second time is to make sure you’ve typed it correctly. Use a mixture of upper and lower case, numbers and symbols to make it more secure. Click the Next button

6. Enter your date of birth, gender, country and email address. This personal information is used to help you recover your password if you forget it, and is used by the Scratch team to understand who uses Scratch. It doesn’t appear on the website anywhere. Click the Next button

7. Click OK Let’s Go! and you will be logged in. Simply click Create on the navigation bar at the top of the screen, and you’re ready to start programming!

The design of websites can change from time to time, so don’t worry if you see variations in the sign-up process when you do it.

You can try Scratch 2.0 by just going to the website and clicking Create at the top of the screen. If you use an account, though, the website will automatically save your work for you.

When you return to the site next time, you can just click “Sign in” in the top right to get to all your projects.
Introducing Scratch

Using the Scratch screen

To start using Scratch 2.0, visit http://scratch.mit.edu/ in your web browser and click Create at the top of the screen. To start using Scratch 1.4, double-click its icon on your desktop.

The most obvious difference between the two versions of Scratch is the way the screen is laid out. With the exception of the Backpack (a new feature in Scratch 2.0), all the same elements are there, but they’ve been juggled around a bit. The background colors have been lightened in Scratch 2.0 too. This page shows the screen layout in Scratch 2.0, and the facing page shows the screen layout in Scratch 1.4.

The main parts of the screen are:

- **Stage**: This is where you can see your animations and games in action. When Scratch starts, there’s a large orange cat in the middle of the Stage. In Scratch 2.0, the Stage is on the left, whereas in Scratch 1.4, the Stage is on the right.

- **Sprite List**: The cat is a ‘sprite’, which is like a character or object in a game. Your project might include lots of sprites, such as the player’s spaceship, invading aliens and a missile. In the Sprite List, you can see all the sprites that are in your project, and click them to switch between them. In both versions of Scratch, the Sprite List is underneath the Stage.

Don’t forget

If you’re using a Raspberry Pi, you’re using Scratch 1.4. If you’re using Scratch in a browser, you’re using Scratch 2.0.

Hot tip

Use the tabs above the Blocks Palette (in Scratch 2.0) or above the Scripts Area (in Scratch 1.4) to switch between the scripts, costumes and sounds on a sprite. You’ll learn all about how to use these later!
Blocks Palette: In Scratch, you give the computer commands by using blocks, which are instructions that fit together like jigsaw pieces. The Blocks Palette presents you with all the blocks you can use. When you start Scratch, you can see the Motion blocks, which are color-coded in dark blue, and are used for moving sprites around the Stage. You can browse a different set of blocks in the Blocks Palette by clicking one of the buttons above it, such as the Looks button or the Sound button.

Scripts Area: The Scripts Area is where you make your programs in Scratch, by assembling blocks there. This area expands to fill the screen space available, so if you use a larger monitor, the Scripts Area will be bigger.

Backpack: The Backpack is a new feature in Scratch 2.0, which you can find underneath the Blocks Palette and Scripts Area. Click it to open it. It works a bit like a clipboard. You can copy scripts or sprites to it by dragging them there and dropping them. If you want to use them, just drag them from the Backpack back into your project. Your Backpack works across all your projects, so it's a great way to copy sprites or bits of program between different projects.

Don’t forget

You’ll see all these elements in action soon, so don’t worry about memorizing the screen layout. This section is just to help you get your bearings. Remember that these pages are here to refer to at any time.
Exploring the blocks

Before we start making a program, try experimenting with a few blocks to see what they do:

1. Click the *Motion* button above the Blocks Palette to show the Motion blocks. This button is selected when you first start Scratch.

2. In the Blocks Palette, click the *move 10 steps* block. The cat on the Stage moves in the direction it’s facing, to the right. Each time you click the block, the cat moves once. This block only changes the cat’s position, though: you won’t see its legs move.

3. The number of steps is how far across the screen you want the cat to move. Click the number 10 and change it to something else. Try 50 and when you click the block, the cat moves five times as far. Whenever you see a white hole in a block, you can change what’s in it.

4. Rotate the cat by clicking the *turn clockwise 15 degrees* block. To change the angle of the turn, change the number. Remember to click the block to actually make the cat turn. When you click the *move 10 steps* block next time, the cat walks in its new direction.

---

**Hot tip**

Sometimes I’ll show you what something looks like in both Scratch 2.0 and Scratch 1.4. The picture on the left, or top, will be Scratch 2.0. The one on the right, or bottom, will be Scratch 1.4.

**Hot tip**

Scratch won’t let the cat disappear off the screen completely. If you use numbers that are too big, the sprite will stay at the edge of the screen.

Above: The cat on the Stage, after I clicked Turn Clockwise 15 Degrees.
5. If the cat gets to the edge of the Stage, drag it back again with your mouse pointer. Click the cat, hold the mouse button down, move the cat, and then release the mouse button to drop it in place.

6. Click the **Pen** button above the Blocks Palette.

7. Click the **pen down** block. Now, if you click the Motion button and click the blocks to move the cat around, it will leave a line behind it wherever it goes. There is also a **pen up** block you can use to turn this effect off again.

8. If you’re using Scratch 2.0, click the **Looks** button above the Blocks Palette.

9. In Scratch 2.0, click the **next costume** block to see the cat’s legs move, so it appears to run on the spot. Costumes are just different pictures a sprite can have, and the cat has two that show its legs in different positions (see below).

---

**Hot tip**

All these blocks are explained later in the book, but for now it’s worth spending a few minutes exploring some of the commands you can give the cat.

**Hot tip**

In Scratch 1.4, the cat only has one costume, but I’ll show you how to add additional costumes in Chapter 5.
Introducing Scratch

Before we make our first program, let’s change the background of the Stage to something more inspiring. The way you do this is different in Scratch 2.0 and Scratch 1.4.

Choosing a backdrop in Scratch 2.0

1. To the left of the Sprites List, there is a panel for the Stage. Underneath the heading New Backdrop, click the first icon to choose a backdrop from the library. The other icons enable you to paint a backdrop, upload a picture from your computer, or use your webcam to take a photo.

2. When the library opens, click the themes and categories on the left to view different backdrops available, and use the scrollbar on the right to see more designs. Click the Nature theme.

3. Click the hill image, and then click the OK button.

4. Your backdrop is added to the Stage, behind the cat, and the Paint Editor opens on the right so you can edit the background if you want to. We’ll look at the Paint Editor in Chapter 4.
Choosing a background in Scratch 1.4

1. To the left of the Sprite List is a white icon that represents your blank Stage. Click it to start.

2. Click the **Backgrounds** tab above the Scripts Area, then click the **Import** button.

3. Use the file browser to explore the backgrounds available. Double-click a folder icon to look inside that folder. To go back to the previous folder, click the **up arrow** indicated below.

4. Go into the Nature folder and use the scrollbar on the right to find the hill picture. Click it and then click **OK**.

5. Your background is added to the Stage, behind the cat.

---

You can also use the file browser in Scratch 1.4 to find your own files on your computer to use as backgrounds. In the left of the file browser, Click Computer, your username or Desktop to start browsing your computer. To get back to the Scratch backgrounds again, click Backgrounds.
**Introducing Scratch**

When you click blocks in the Blocks Palette, the cat moves immediately, so this is good for testing what blocks do, but not useful for making a program. A program is a set of repeatable instructions that you can store up to carry out later. For our first Scratch program, let’s make the cat walk down the hill:

1. We’re going to write a program for the cat so click the cat in the Sprite List.

2. Next, check that the Scripts Area is open. If the Scripts Area is empty, you’re ready. If it shows costumes or sounds instead, click the **Scripts** tab. In Scratch 2.0, the **Scripts** tab is above the Blocks Palette. In Scratch 1.4, it’s above the Scripts Area.

3. Click the **Motion** button above the Blocks Palette.

4. You make a program by dragging blocks into the Scripts Area from the Blocks Palette. To do this for your first block, click the **turn clockwise 15 degrees** block in the Blocks Palette, hold the mouse button down, move the mouse pointer into the Scripts Area and then release the mouse button. This first block will point our cat downhill, ready for its walk.

5. Click the **move 10 steps** block in the Blocks Palette, drag it into the Scripts Area and drop it underneath the **turn clockwise 15 degrees** block. They will snap together. When blocks are joined like this they make what’s known as a “script”. A sprite can have more than one script, and a program might include lots of sprites with several scripts.

---

**Creating your first program**

---

**Hot tip**

If you click and drag a block that’s joined to other blocks in the Script Area, it will break away from them, and carry all the blocks underneath it with it.

**Beware**

You can move blocks around the Scripts Area, but if you drag them into the Blocks Palette, they’ll be deleted.

**Don’t forget**

Make sure the blocks snap together, otherwise they won’t work as one script. If they don’t snap together, they’re not close enough.
Click the Control button above the Blocks Palette. Control blocks are used to decide when things should happen. Drag the wait 1 secs block into the Scripts Area and snap it underneath the other two blocks. This block adds a 1 second delay. Without it, our cat will move so fast, it’ll appear to just jump from the start of his walk to the end. Slowing him down enables us to see what’s going on. You can make him walk a bit faster by changing the delay from 1 second to 0.5 seconds.

Right-click the move 10 steps block, and when the menu opens, choose Duplicate. This copies the block plus any blocks underneath it in your script. In our example, it copies the move and the wait blocks. Move the copy to the bottom of your program, and click to place the blocks there. You can repeat this step several times to make the cat walk further.

It’s a long walk for a tiny cat, so let’s make him finish his walk with an exclamation of ‘Phew!’ in a speech bubble. Click the Looks button above the Blocks Palette, and drag the say Hello! for 2 secs block into the Scripts Area and join it to your program. Click Hello! to edit what the cat says to Phew!
When you start a script’s commands, it’s called “running” the script. To run your script, click any of the joined-up blocks in the Scripts Area. Scratch carries out all the joined-up instructions in order, starting at the top and working its way down the blocks.

What happens if you click the script to run it again? The cat turns again and walks from where it finished last time. Eventually, it’ll be walking on its head. Let’s add some blocks to put it in the right starting position. Click the Motion button above the Blocks Palette and drag in the point in direction 90° block and the go to x:0 y:0 blocks. If the go to block has different numbers in it, edit them both to make them zero. Add these blocks to the top of your script.

Experiment! If you use a negative number of steps in the “move 10 steps” block, the sprite moves backwards. Why not see if you can make the cat walk backwards up the hill again?
Saving your project

A Scratch project includes all the sprites, scripts and backgrounds that are used in it. It’s a good idea to save your projects so you can come back to them later to reuse them or modify them.

**Saving projects in Scratch 2.0**

In Scratch 2.0 (web version), your work is automatically saved for you as you make changes to your project. In the top-right corner of the screen, you can see whether your latest changes have been saved. If they haven’t, there will be a link here to **Save Now**.

Your project is saved with the name Untitled plus a number. You can choose a more useful name by editing the box above the Stage.

There are additional options for saving your work in the File menu, also above the Stage, see below. These include:

- **Save as a copy**: This makes a copy of your project with a new name. The previously saved version of your project is left untouched. Use this if you want to experiment with your program without losing a working version of it.

- **Download to your computer**: This enables you to save your project as a file on your computer. You can open it using the downloadable version of Scratch 2.0. If your Internet connection fails, use this option straight away to save your work!

- **Upload from your computer**: If you previously downloaded a Scratch project to your PC, or used a downloadable version of Scratch to create it, use this option to upload it to the Scratch 2.0 website.

- **Revert**: This throws away all the changes you’ve made to the project since you opened it this time.

---

**Beware**

You can’t use projects made using Scratch 2.0 in Scratch 1.4. You can use Scratch 1.4 projects in 2.0, though.

---

**Hot tip**

In the downloadable version of Scratch 2.0, your work isn’t automatically saved for you. Use the **Save** option in the File menu to save your work. Use the **Save As** option to save your project with a new filename, so you don’t overwrite the previous version.
Introducing Scratch

...cont’d

Saving and opening your project in Scratch 1.4

Click the File menu above the Blocks Palette at the top of the screen. This menu’s options include:

- **Open**: Opens a previously saved file.

- **Save**: Use this to save a new project, or an old project you’ve been editing. If you’re saving a new version of a saved project, you’ll replace the old saved version.

- **Save as a copy**: If you want to keep the previous version of a project, use this option. It will save a new file containing your project, and leave the previous file untouched.

- **Import Project**: This enables you to open a project, and combine its sprites, scripts and backgrounds with the project that’s currently open.

- **Export Sprite**: This option enables you to save a sprite as a costume you can use in other projects. You’ll learn about costumes in Chapter 5.

When you save a new project or a new copy of a project, the file browser opens, as shown below. The buttons down the left are used to choose where to save your file. Click the folder name at the top (Scratch Projects in the screenshot) to choose another drive or folder, and click the up arrow beside it to go up a folder. Type the filename in the box at the bottom and click **OK**.

*Don’t forget*

If you are using Scratch 1.4, remember to save regularly to make sure you don’t lose any of your work.
Opening projects

In the downloadable version of Scratch 2.0, you open projects through the File menu at the top of the screen. The website is a bit more complicated. To find your projects there, click your username in the top-right of the screen and then click My Stuff. If it says “Sign in” in place of your username, click it to sign in first. The My Stuff section shows all your projects, with those you most recently edited nearer the top. Take a look at my projects in the screen below:

![Scratch website interface showing My Stuff section]

Use the scrollbar at the right edge of your browser window to find more projects and click the Load More button when you get to the bottom of the list. If no more projects load when you click the button, that’s all of them. To open a project, click its See inside button.

Because projects are saved automatically, your My Stuff area quickly fills up with Untitled projects. To tidy up, delete unwanted projects by clicking their Delete links on the right. If you delete a project by mistake, click the Trash folder on the left, and then click the Put back button to recover the project.

Beware

If you click the Share button on a project, it will be available for anyone to see, use and reuse. See Chapter 11 for more on sharing your projects.

Don’t forget

In Scratch 1.4, you open projects by clicking to open the File menu at the top of the screen and then choosing Open.
Opening shared projects

You can open the projects that other people have shared on the Scratch website too. Visit the website at http://scratch.mit.edu and click Explore at the top of the screen. Use the options on the left to choose a category.

The menus in the top right enable you to sort by the most loved, most remixed and most recent, and to choose how new you want the projects to be.

Click a project to go to its page. The instructions on the right tell you how to use the project, and you click the green flag button in the middle of the player to run it. If you like what you see, and you want to know how it was done, click the See inside button in the top right to go into the editor and see the scripts, sprites and backgrounds that make it work.

Beware
At the time of writing, there isn’t a way to download shared projects from the website to use in Scratch 1.4.
A

A, B, C, D sockets 184
adding sprites 52–53, 93
Adobe Flash Player 11, 180
alphabet sprite 140
animation 30, 54–55, 95, 134
asking questions 125
average 132

B

backdrops. See backgrounds
background music 169, 177
backgrounds 18–19, 51, 110
changing 110, 111
Backgrounds tab 19
Backpack 15
bats 182
bitmaps 52, 85
blocking paths 201
blocks 15, 16
color of 41
Control 21
broadcast [message] 103, 173
broadcast [message] and wait
103–104, 106, 146, 174
create clone of myself 76
delete this clone 76
forever 41, 55, 88–89, 175
forever if 207
if 68, 89, 106, 143
if... else 61
if... then... else 61–62, 130–131, 164
repeat 10
39–40, 49, 114, 127, 155, 161, 208
repeat until 58–60, 121–123, 147, 156, 159–160
say Hello! for 2 secs 164
stop all 79, 164, 177, 207
stop script 207
wait 1 secs 21, 33, 55, 73, 87, 204
wait until 88
when green flag clicked 45, 48, 51
when I receive [message] 103, 108, 111, 142, 152
when I start as a clone 76–77
when space key pressed 56
when Sprite1 clicked 203
Data
add thing to [list name] 150, 152, 154
change [variable name] by 1 47, 50, 88, 162
delete 1 of [list name] 150, 152
hide list [list name] 151
hide variable [variable name] 47
insert thing at 1 of [list name] 150
item 1 of [list name] 151, 153, 156
length of [list name] 151, 197
[list name] contains thing 151, 160
replace item 1 of [list name] with thing 150, 162
set [variable name] to 0 47, 67, 154
show list [list name] 151
show variable [variable name] 47

Events
broadcast [message] 103, 173
broadcast [message] and wait
103–104, 106, 146, 174
when backdrop switches to [backdrop name] 111
when green flag clicked 45, 48, 51
when I receive [message] 103, 108, 111, 142, 152
when loudness > 10 203
when space key pressed 56
when this sprite clicked 203

Looks
15, 17, 21
change color effect by 25 144, 172
clear graphic effects 144, 171
costume # 143
go back 1 layers 86
go to front 73, 86
hide 41, 48, 77, 203
next costume 134, 142, 183
say Hello! 127
say Hello! for 2 secs 21, 62
set color effect to 0 51, 78, 144, 167, 172
set size to 100% 54, 67, 79, 90
show 73, 203
switch costume to [costume name] 134
making 199

More Blocks
define 200

Motion
15, 16, 30, 38
change x by 10 30, 55, 69
change y by 10 30, 69
glide 1 secs to x:0 y:0 30, 202
go to [mouse-pointer] 42
go to [Sprite1] 185
go to x:0 y:0 22, 28, 29, 33, 72
if on edge, bounce 73, 183, 193, 194
move 10 steps 16, 20, 22, 35, 38–39, 50, 90, 174
point in direction 90 22, 35, 72, 123, 173
point towards 35
set rotation style 37, 175
set x to 0 30
set y to 0 30
turn anti-clockwise 15 degrees 35
turn clockwise 15 degrees 16, 20, 35, 38–39
Operators
- 188
* 123, 188
/ 132
+ 187
< 78, 195
= 143, 156, 159, 162
> 89, 178
and 59
join hello world 126, 128–129, 178
length of world 153, 159
letter 1 of world 162
not 198
or 59, 147
pick random 1 to 10 72, 86, 128

Pen 17, 31–33, 38, 48, 192–193, 202
change pen color by 10 32, 40, 42, 202
change pen shade by 10 32, 202
change pen size by 1 32
clear 31, 33, 146
pen down 17, 31, 33
pen up 31, 33
set pen color to 0 32
set pen color to [color] 31, 33, 193
set pen shade to 50 32
set pen size to 1 32, 33, 193
stamp 32, 157, 198

Sensing
answer 125–126
ask [What’s your name?] and wait 125–126, 129, 197
key space pressed? 68, 127
loudness 180, 185
reset timer 121–123
sensor button pressed? 185, 187–188
set video transparency to 50% 180, 183
slider sensor value 185, 187
timer 119, 121–123, 132
touching? 60, 74, 193
touching color? 60–62, 193, 201
turn video on 180, 183
username 178
video motion on this sprite 180, 183
x position of Sprite1 185
x position of [sprite name] 195
y position of Sprite1 185

Sound
15, 101, 107
change tempo by 20 107
change volume by -10 101
play drum 1 for 0.2 beats 107
play note 60 for 0.5 beats 107–108, 112–114, 124
play sound [sound name] 177
play sound [soundname] 101
play sound [soundname] until done 101, 105
rest for 0.2 beats 107
set instrument to 1 107
set tempo to 60 bpm 107
set volume to 100% 101
stop all sounds 101, 177
tempo 107
volume 101

Variables
47
add thing to [list name] 150, 152, 154
change [variable name] by 1 47, 50, 88, 162
delete 1 of [list name] 150, 152
hide list [list name] 151
hide variable [variable name] 47, 67
insert thing at 1 of [list name] 150
item 1 of [list name] 151, 153, 156
length of [list name] 151, 197
[list name] contains thing 151, 160
replace item 1 of [list name] with thing 150, 162
set [variable name] to 0 47, 67, 154
show list [list name] 151
show variable [variable name] 47

Blocks Palette
14–15, 20–21, 41
bouncing 73, 193
bracket 62
brightness 144, 186
broadcasts 103, 146, 198–199
browser 8, 11, 13
bugs 206–208
button 184

cars, moving 194
cat 14, 17, 33
chat program 197
checking for movement 183
checking input 159
cleanup 206
clipboard 15
cloning sprites. See sprites, cloning
Cloud Data Log 170
cloud variables 47, 170, 178
collision detection 62, 74, 88, 176, 194
color 31
  graphic effect 144
  in Paint Editor 84
comments 206, 209
community 8, 209
Control. See blocks, Control
coordinates 28, 30, 68
copying blocks. See duplicating scripts
copying costumes. See duplicating costumes
copying scripts. See duplicating scripts
copying sprites. See duplicating sprites
costumes 17, 20, 198
  adding 93–95
  copying. See duplicating costumes
drawing 138–139
Costumes Area 93–95, 138
Costumes tab 80, 93–95
Create 13, 14
cursor keys 57, 68

D

dance 133
Data. See blocks, Data
death sequence 173
debugging 206–208
deleting
  lists 149
  projects 25
  scripts 69
  sprites 75
  variables 149
detecting
  sprite position 195
  walls 201
difficulty, adjusting 90, 171, 204
direction 35, 71
division 132
download shared projects 26
Download to your computer 23
dragging blocks 20
drawing
  a clock 119–120
  a hexagon 39–40
  a house 33
  a square 34, 38–39
  a triangle 34
  on the Stage 31
  patterns 40
  using directions 38
duplicating
  backgrounds 110
  costumes 138, 168
  scripts 15, 21, 61, 69, 106
    between sprites 86
  sprites 15, 75, 76, 105

e
electronics 189
energy meter 78
errors 10, 206–208
Events. See blocks, Events

E

examples
  Abstract Artist 202
  blinking LED 189
  clap-o-meter 185
  Cosmic Chorus 92
  Donut Chaser 181
downloading 44
drawing a house 33
drawing a spiral 46, 48
  Going Batty 182
  Hangman 136
  hexagons pattern 40
  Keepy-Uppy 192
  London Bridge 114
  Maze Mania 201
  Night Flight 186
  on Scratch website 44
  opening in Scratch 1.4 24
  Penguin Patter 197
  Photo Safari 203
  Quiz Break 116
  Rainbow Painter 41–42
  Ransom Writer 198
  Shop Cat 73, 194
  singing Gobo 102
  singing sprite 108
  Space Swarm 166
  Spiral Rider 44
  Super Dodgeball 64, 194
teleport 172
  victory dance 133
  walking cat 20
  excited dinosaur 147
  Explore 209
  Export Sprite 24

F

feedback 209
file browser 19, 24, 53
File menu
  Scratch 1.4 24
  Scratch 2.0 23
firing 174, 176
fisheye 144
Flash blocks 207
flying 183, 187
folder 19, 24
football 192–193
Index

G
Game Over 79, 164, 177, 204
General Purpose Input/Output 189–190
ghost effect 51, 78, 144–145, 173
Google Chrome 11, 13
GPIO 189–190
graphic effects 51, 144
gravity 188
green flag 26, 45, 50, 78
grid reference 28

H
hat blocks 45, 56, 76, 108
hide a sprite 41
high score 170, 178

I
importing
backgrounds 19, 110
costumes 95, 140
projects 24
information (of sprites) 37
insert blocks 22
Internet 11
iPad 11

J
joining text 126
Join Scratch 13
jump 187

K
keyboard control 56–57, 68–69, 172, 192–193, 195

L
LED 189
left-right 36, 175
light emitting diode (LED) 189
light sensor 184, 187, 188
limiting variables 89
lists 149–150, 197
deleting 149
lives 171
Looks. See blocks, Looks

loops
  different verses 114
loudness 203
lower case 156

M
Make a Block 199
Make a List 149, 151
Make a Variable 46
maximum value 89
maze 201
Maze Generator 201
menus in blocks 47
microphone 180, 185
missile 168, 174, 176
MIT Media Lab 7, 210
More Blocks 199
See also blocks, More Blocks
Morse code 190
mosaic effect 144, 172
Motion. See blocks, Motion
mouse pointer 42
moving sprites. See blocks, Motion
multiplication 123, 128
music 107, 112–114
  background 169, 177
  beats 113
  notes 112
  rests 113
  scale 109
  sheet music 112
My Stuff 25, 209

N
nested loop 40
new sprite 52–53
number1 200

O
obstacles 194
Open 24
opening projects 23–25, 26
Operators. See blocks, Operators

P
Paint Editor 18, 80–85, 95
tools 82–83
vectors 85
paint new sprite 80–81
password 13, 209
Pen. See blocks, Pen
photo 52
PicoBoard 184–188, 210
pipette 31
pixelate 144
Print Screen key 204
privacy 170, 178, 180
program 10, 20
structuring 146–148
programming 7, 10
programming language 7, 10
project 20
deleting in Scratch 2.0 25
opening 24–25
saving 23–24
sharing 209
Put back 25

Q
questions, asking 125

R
random 72, 74, 176, 183, 193
direction 72, 128, 167
list items 197
numbers 72, 202
pattern 87, 203
pause 72, 74, 86
screen position 183
Raspberry Pi 11, 12, 66, 67, 79,
101, 107, 166, 169, 177, 186, 189–190, 210
pins 189–190
recovery projects 25
remix 8, 26
removing scripts 69
renaming sprites 71
reporting boxes 66
resistance 184
resistors 189
resizing sprites 75
Resnick, Mitchel 7
Revert 23
right-clicking 21
rotating sprites 36–37
rotation style 36–37, 71, 175
running a script 22
saving 24
as a copy 23, 24
backgrounds 110
projects 23–24
scissors icon 75
score 65, 67, 88, 132, 170, 176, 195, 204
high score 170, 178
Scratch 1.4 11, 14–15, 23, 24, 37, 81, 99
Scratch 2.0 11, 14–15, 18, 23, 25, 37, 76, 80
cloud variables. See cloud variables
sounds 97
vectors 85
Scratchboard Watcher 186
Scratcher 8
ScratchGPIO 190, 210
screen layout 14
scripts 20
creating 20
defined 20
deleting. See deleting scripts
duplicating. See duplicating scripts
Scripts Area 15, 20, 22, 33, 95
Scripts tab 20, 33, 51, 95
See inside 25, 26
Sensing. See blocks, Sensing
Share 25
shared projects 26, 45, 170
Share This Project Online 209
sharing projects 209
Sign in 13, 25
Single Stepping 207
slider 184, 188
Small stage layout 80
soccer 192–193
Sound. See blocks, Sound
sound effects 169, 174, 175, 177
rising 124
typewriter 198
Sound Library 98, 99
sounds 20
adding 97–99, 169
detecting 180
playing 101
recording 98, 100
Sounds Area 97, 100
sound sensor 184
Sounds tab 97
Sprite Library 52–53, 95, 140
<table>
<thead>
<tr>
<th>Index</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprite List</td>
<td>14, 20, 60, 71</td>
</tr>
<tr>
<td>sprites</td>
<td>20</td>
</tr>
<tr>
<td>adding</td>
<td>52–53, 93–95</td>
</tr>
<tr>
<td>cloning</td>
<td>76–77, 182</td>
</tr>
<tr>
<td>costumes. See costumes</td>
<td></td>
</tr>
<tr>
<td>customizing</td>
<td>80</td>
</tr>
<tr>
<td>defined</td>
<td>14</td>
</tr>
<tr>
<td>deleting. See deleting sprites</td>
<td></td>
</tr>
<tr>
<td>detecting position</td>
<td>195</td>
</tr>
<tr>
<td>hiding</td>
<td>203</td>
</tr>
<tr>
<td>random</td>
<td>53</td>
</tr>
<tr>
<td>renaming</td>
<td>71</td>
</tr>
<tr>
<td>resizing</td>
<td>75</td>
</tr>
<tr>
<td>revealing</td>
<td>86</td>
</tr>
<tr>
<td>uploading</td>
<td>52</td>
</tr>
<tr>
<td>Stage</td>
<td>14, 18–19, 28, 31, 51, 123</td>
</tr>
<tr>
<td>reporting boxes</td>
<td>66</td>
</tr>
<tr>
<td>stamp</td>
<td>75</td>
</tr>
<tr>
<td>See also blocks, Pen</td>
<td></td>
</tr>
<tr>
<td>starting a script. See running a script</td>
<td></td>
</tr>
<tr>
<td>stop a program</td>
<td>42</td>
</tr>
<tr>
<td>Stop button</td>
<td>42, 121</td>
</tr>
<tr>
<td>strength</td>
<td>65, 78, 88–89</td>
</tr>
<tr>
<td>string</td>
<td>199</td>
</tr>
<tr>
<td>string1</td>
<td>200</td>
</tr>
<tr>
<td>strings, joining</td>
<td>126</td>
</tr>
<tr>
<td>structuring programs</td>
<td>146–148</td>
</tr>
<tr>
<td>surprise sprite</td>
<td>53</td>
</tr>
<tr>
<td>tags</td>
<td>209</td>
</tr>
<tr>
<td>teleport effect</td>
<td>172</td>
</tr>
<tr>
<td>testing</td>
<td>20, 74, 158</td>
</tr>
<tr>
<td>text, joining</td>
<td>126</td>
</tr>
<tr>
<td>text length</td>
<td>153</td>
</tr>
<tr>
<td>timer</td>
<td>119, 203</td>
</tr>
<tr>
<td>timing</td>
<td>90</td>
</tr>
<tr>
<td>random</td>
<td>87</td>
</tr>
<tr>
<td>title screen</td>
<td>110–111</td>
</tr>
<tr>
<td>transparency</td>
<td>51, 78, 84, 145</td>
</tr>
<tr>
<td>video</td>
<td>180–183</td>
</tr>
<tr>
<td>Turbo mode</td>
<td>207</td>
</tr>
<tr>
<td>sprites</td>
<td>52</td>
</tr>
<tr>
<td>upper case</td>
<td>156</td>
</tr>
<tr>
<td>username</td>
<td>13, 178, 209</td>
</tr>
<tr>
<td>V</td>
<td></td>
</tr>
<tr>
<td>validating input</td>
<td>159</td>
</tr>
<tr>
<td>variables</td>
<td>46–47, 48–49, 65, 114, 125, 141, 152, 170, 206</td>
</tr>
<tr>
<td>cloud</td>
<td>47, 170, 178</td>
</tr>
<tr>
<td>deleting</td>
<td>149</td>
</tr>
<tr>
<td>hiding</td>
<td>118</td>
</tr>
<tr>
<td>in sound effects</td>
<td>124</td>
</tr>
<tr>
<td>limiting</td>
<td>89</td>
</tr>
<tr>
<td>naming</td>
<td>47</td>
</tr>
<tr>
<td>reporting boxes</td>
<td>66, 74</td>
</tr>
<tr>
<td>Variables. See blocks, Variables</td>
<td></td>
</tr>
<tr>
<td>vectors</td>
<td>52, 85</td>
</tr>
<tr>
<td>vehicles</td>
<td>194</td>
</tr>
<tr>
<td>video</td>
<td>180–183</td>
</tr>
<tr>
<td>video direction</td>
<td>181</td>
</tr>
<tr>
<td>video motion</td>
<td>203</td>
</tr>
<tr>
<td>volume</td>
<td>101</td>
</tr>
<tr>
<td>W</td>
<td></td>
</tr>
<tr>
<td>wait, random</td>
<td>87</td>
</tr>
<tr>
<td>walls</td>
<td>201</td>
</tr>
<tr>
<td>web browser. See browser</td>
<td></td>
</tr>
<tr>
<td>webcam</td>
<td>52, 97, 180–183</td>
</tr>
<tr>
<td>website</td>
<td>8, 14, 26, 178, 209, 210</td>
</tr>
<tr>
<td>whirl</td>
<td>144</td>
</tr>
<tr>
<td>writing on the Stage</td>
<td>140, 198</td>
</tr>
<tr>
<td>X</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>28, 30, 68</td>
</tr>
<tr>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>28, 30, 68</td>
</tr>
<tr>
<td>Z</td>
<td></td>
</tr>
<tr>
<td>zig-zag</td>
<td>175</td>
</tr>
</tbody>
</table>
Why choose this book?

It’s written in plain English
Put off by the jargon? Don’t worry, we keep things straightforward.

It’s easy to follow

It’s fully illustrated
We don’t just tell you how to do it, we also show you how.

It’s in full color
This book’s printed in color to make it simpler to use and easier on the eye.

And it’s fantastic value
Do we need to say any more?

£10.99 UK / $14.99 US / $16.95 CN

Scratch is a programming language that is widely used on the Raspberry Pi and in schools and colleges. Scratch’s highly visual interface and drag-and-drop commands make it an ideal language for all ages to try to program. With Scratch Programming in easy steps at hand, learning programming will be a breeze.

This primer introduces you to Scratch fundamentals and then walks you through the commands to create games and animations. Learn to create games that require skill, knowledge or quick fingers, such as Spiral Rider, Space Swarm, or the classic Hangman game. Add music and special effects to your games, and of course keep score.

By the time you’ve finished, you’ll be impressing your friends and family with your own computer games!

Look inside

Enabling the player to fire
To enable the player to fire, we’ll add a script to the player sprite that will broadcast '/fire' when the player presses space. We won’t want the script to fire when the player presses space a second time, so we’ll put an else script to prevent a second press.

Moving and shooting aliens
The alien uses the pixelate effect to make it easier for the player to see the aliens moving across the Stage by pressing a random position on the Stage. You can choose the alien from the Blocks palette, changing direction, making it move forever or stop when the player presses spacebar.

Let these icons make it easier

Wherever you see one of these icons you know there’s a handy tip to spice up your learning, flag something to remember or ward you away from potential dangers.

I was extremely happy with the In Easy Steps series and have purchased several of their guides. Their “get it done” tutorials provide quick and easy reference material for when I’m programming or designing.

Lowter - Webmaster ezine

www.ineasysteps.com