

The Essential Coding Manual

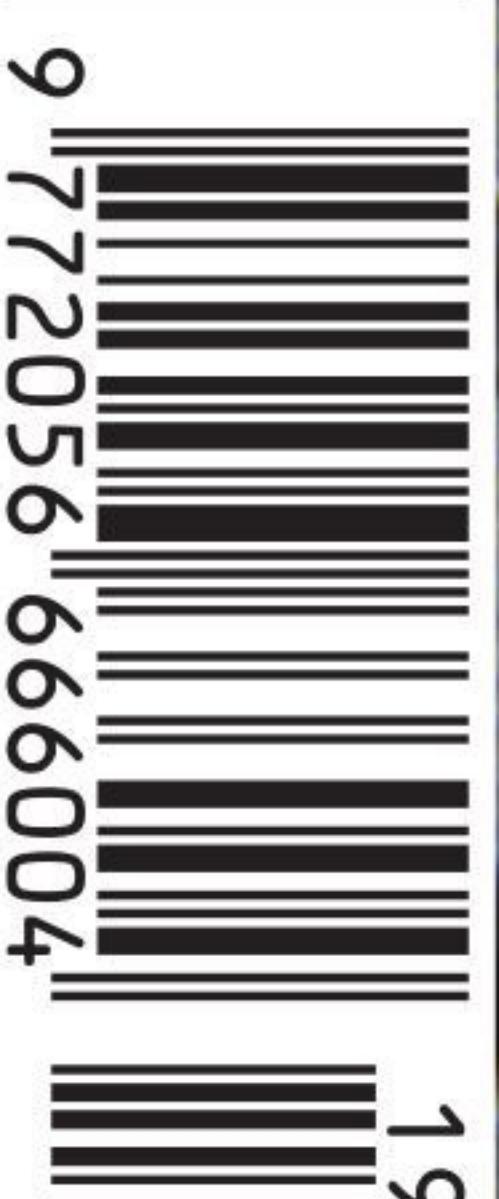
✓ Step-by-step Guides ✓ Coding Projects ✓ Master Coding

Includes
Raspberry
Pi 4



£12.99
Volume 19
US\$24.99
Can\$25.99
Aus\$25.25
NZ\$29.99

100% INDEPENDENT



Inside...:

**190
PAGES**

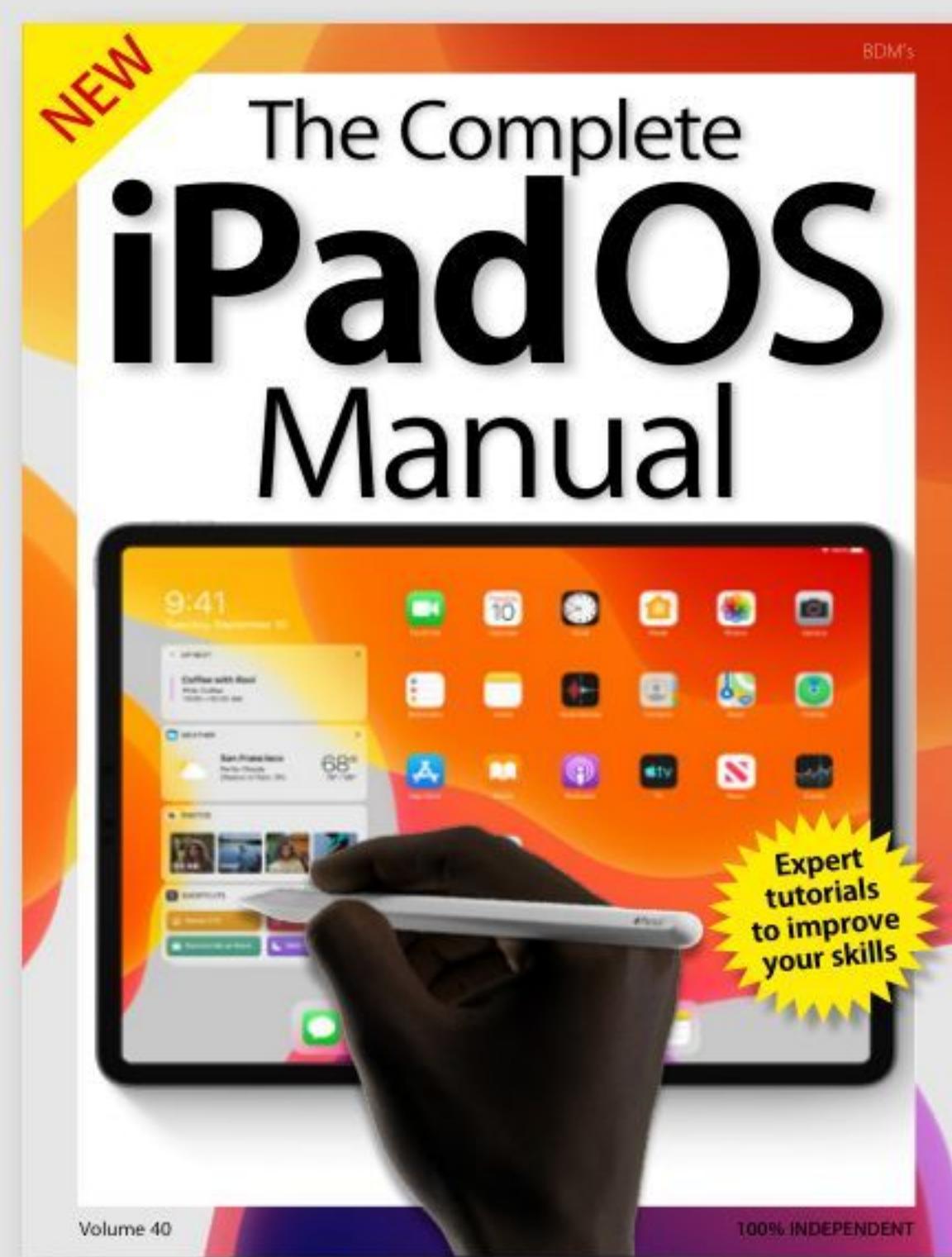
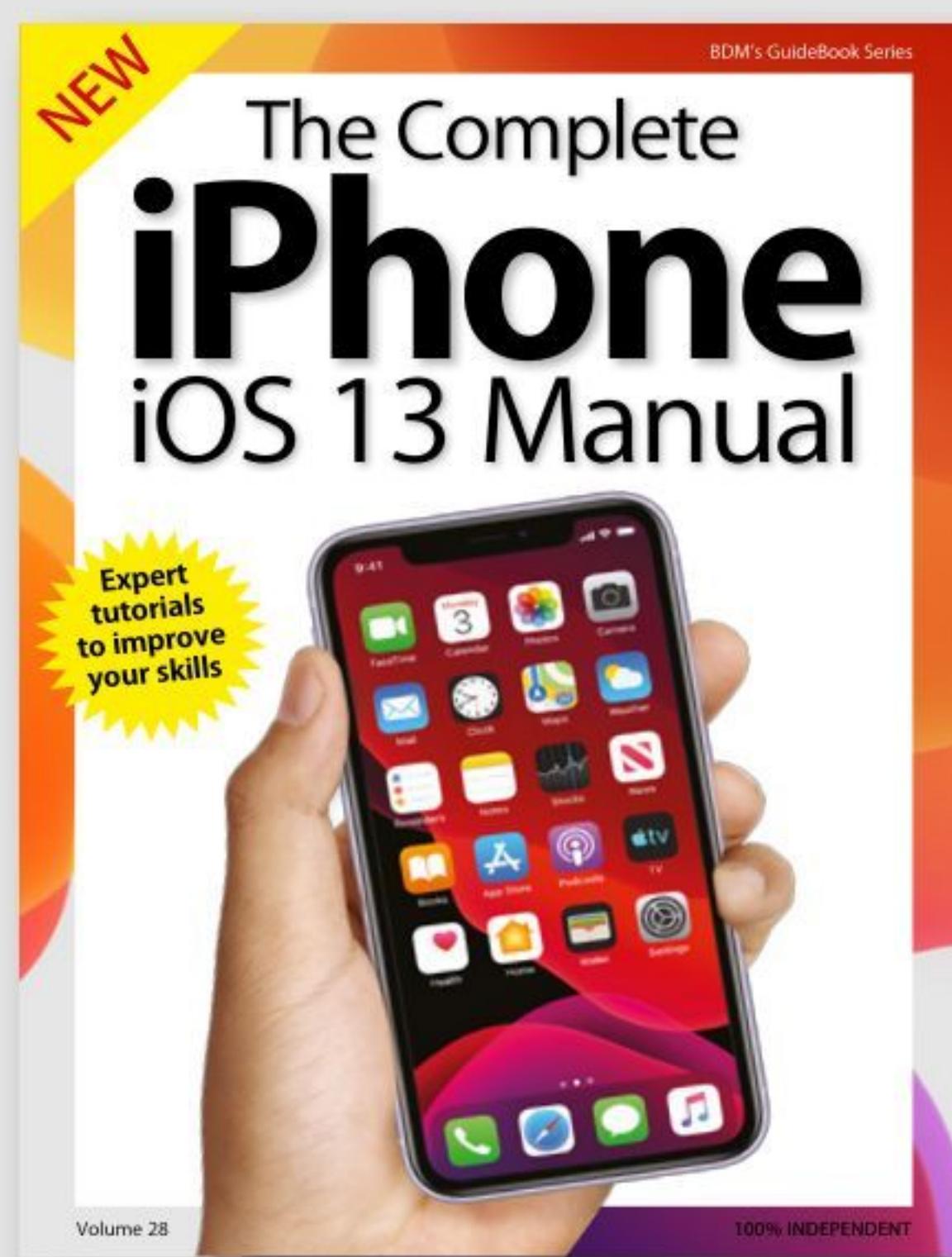
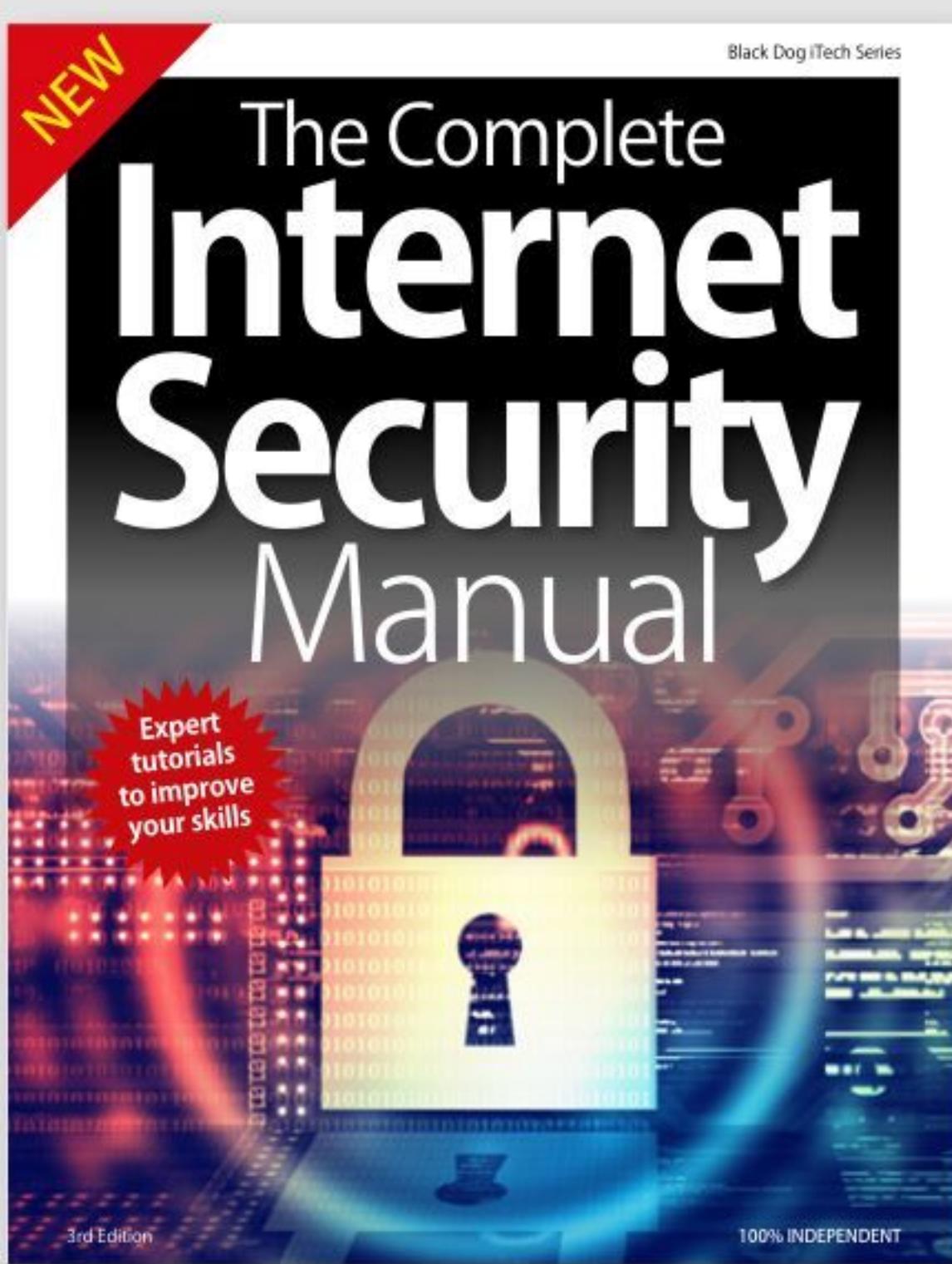
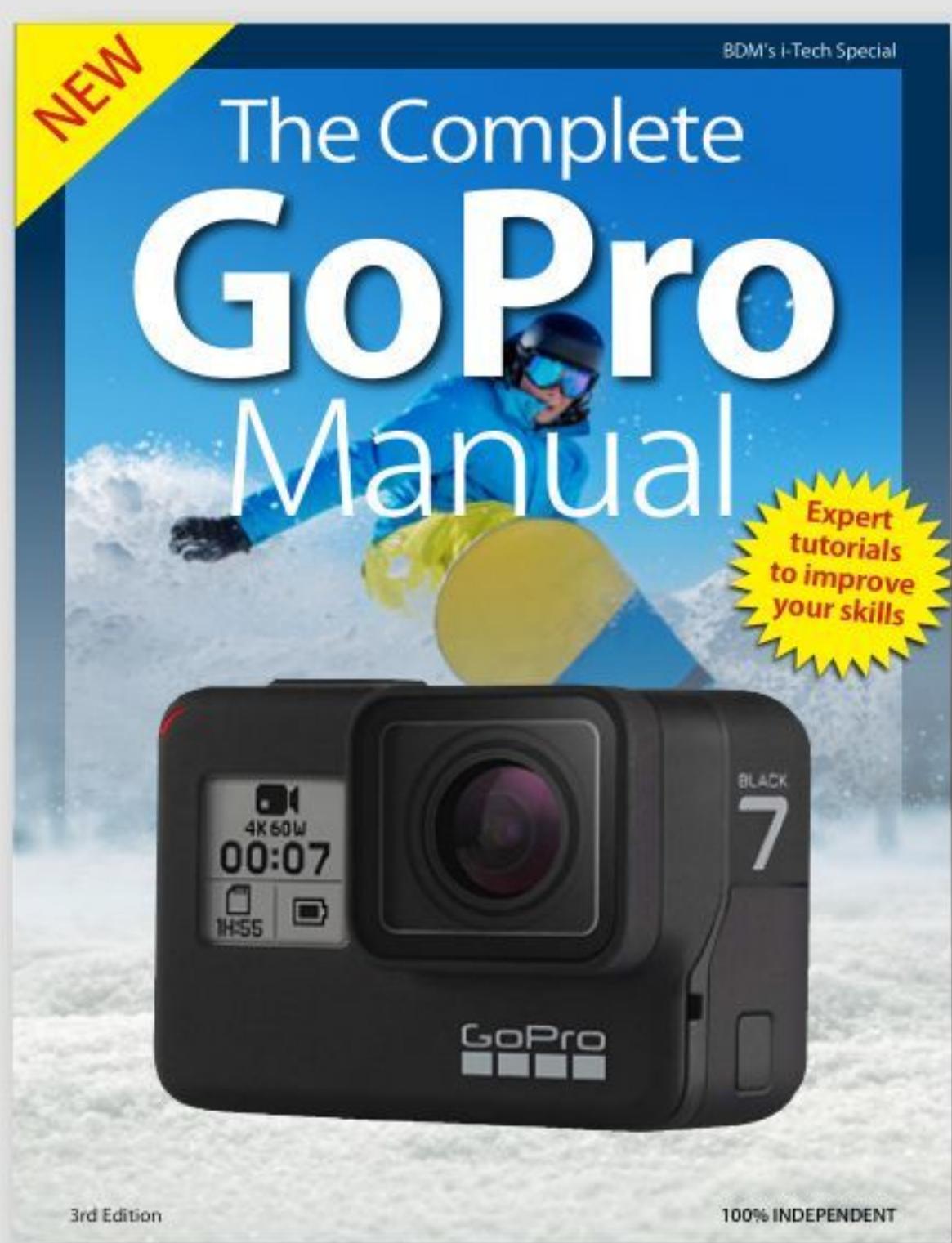
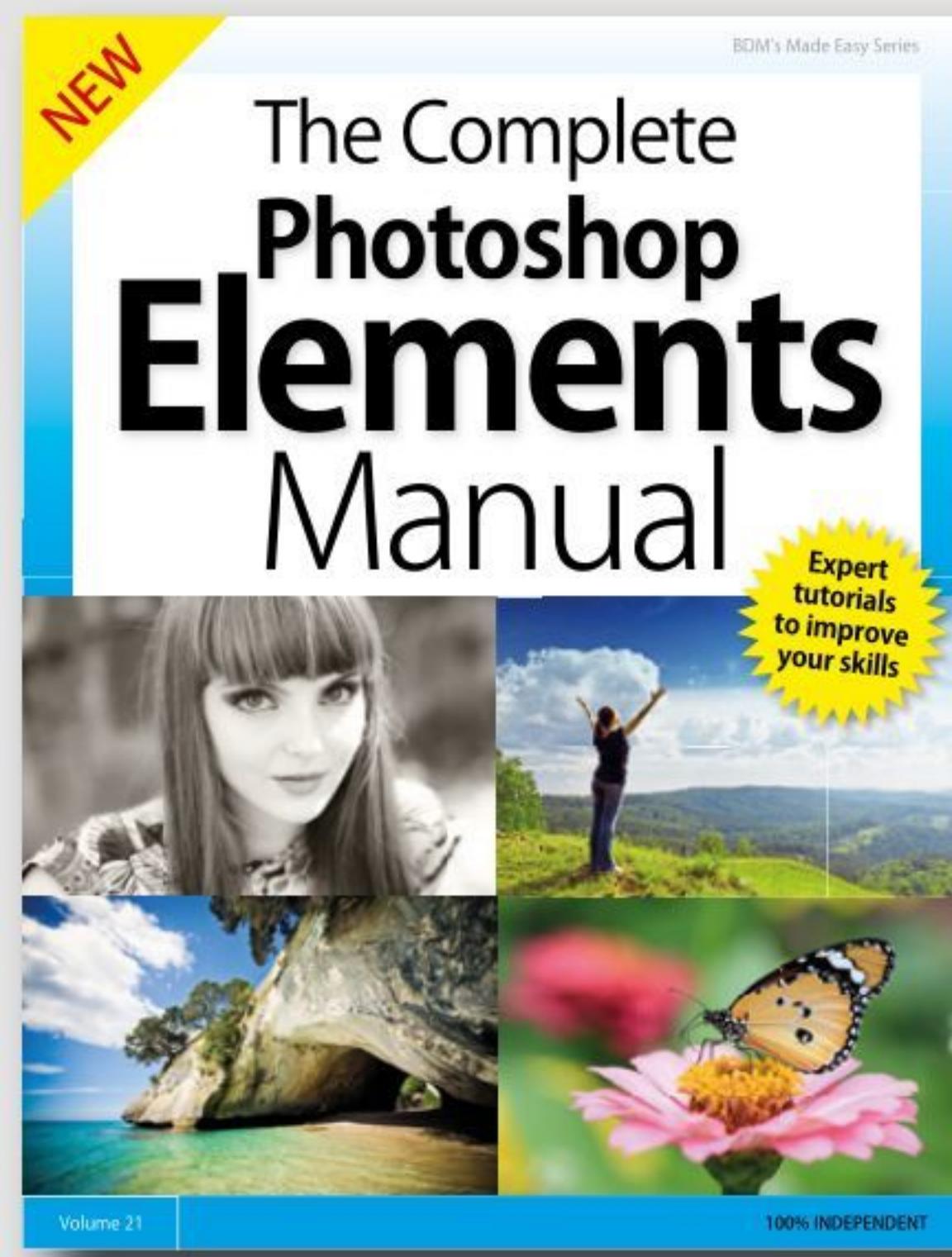
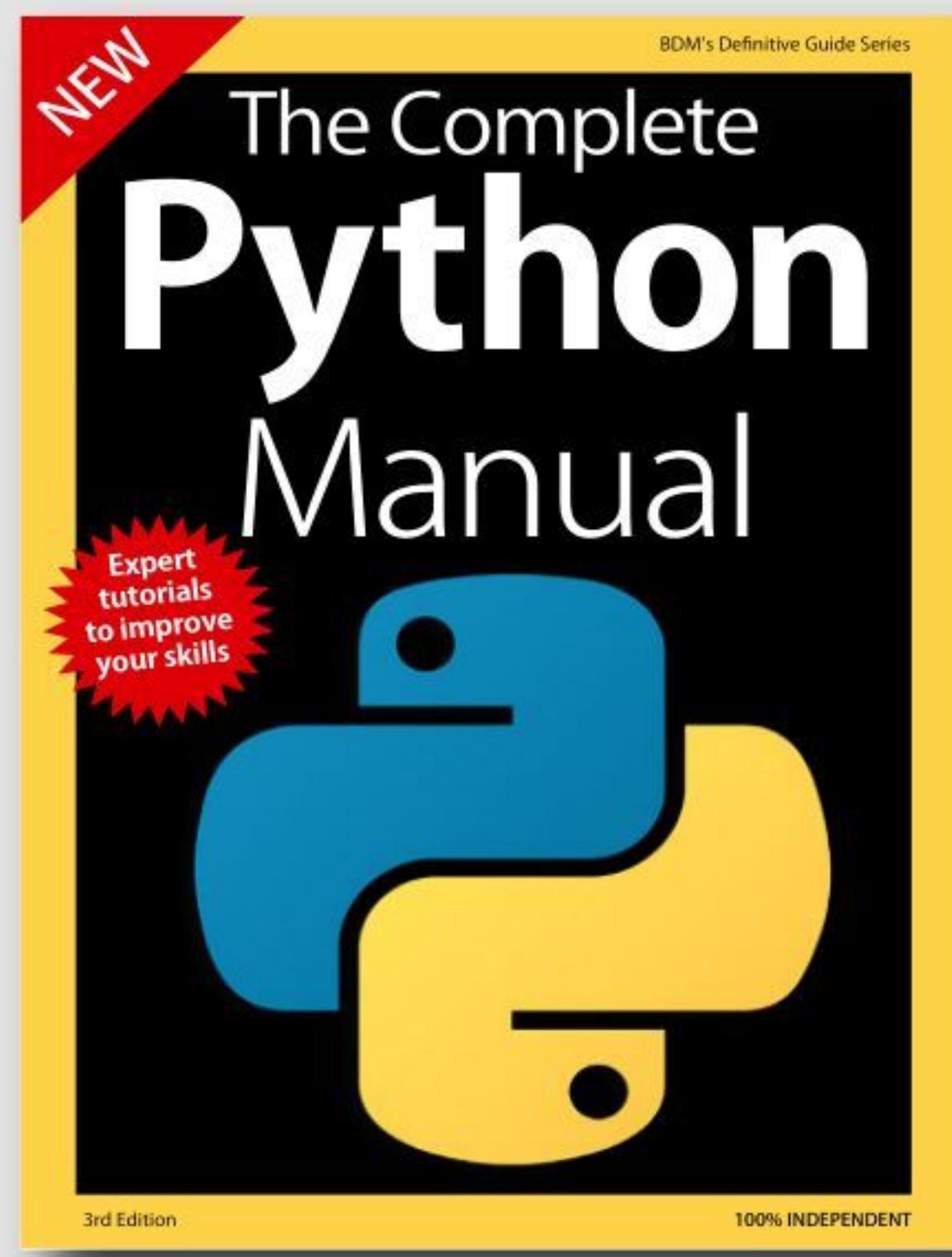
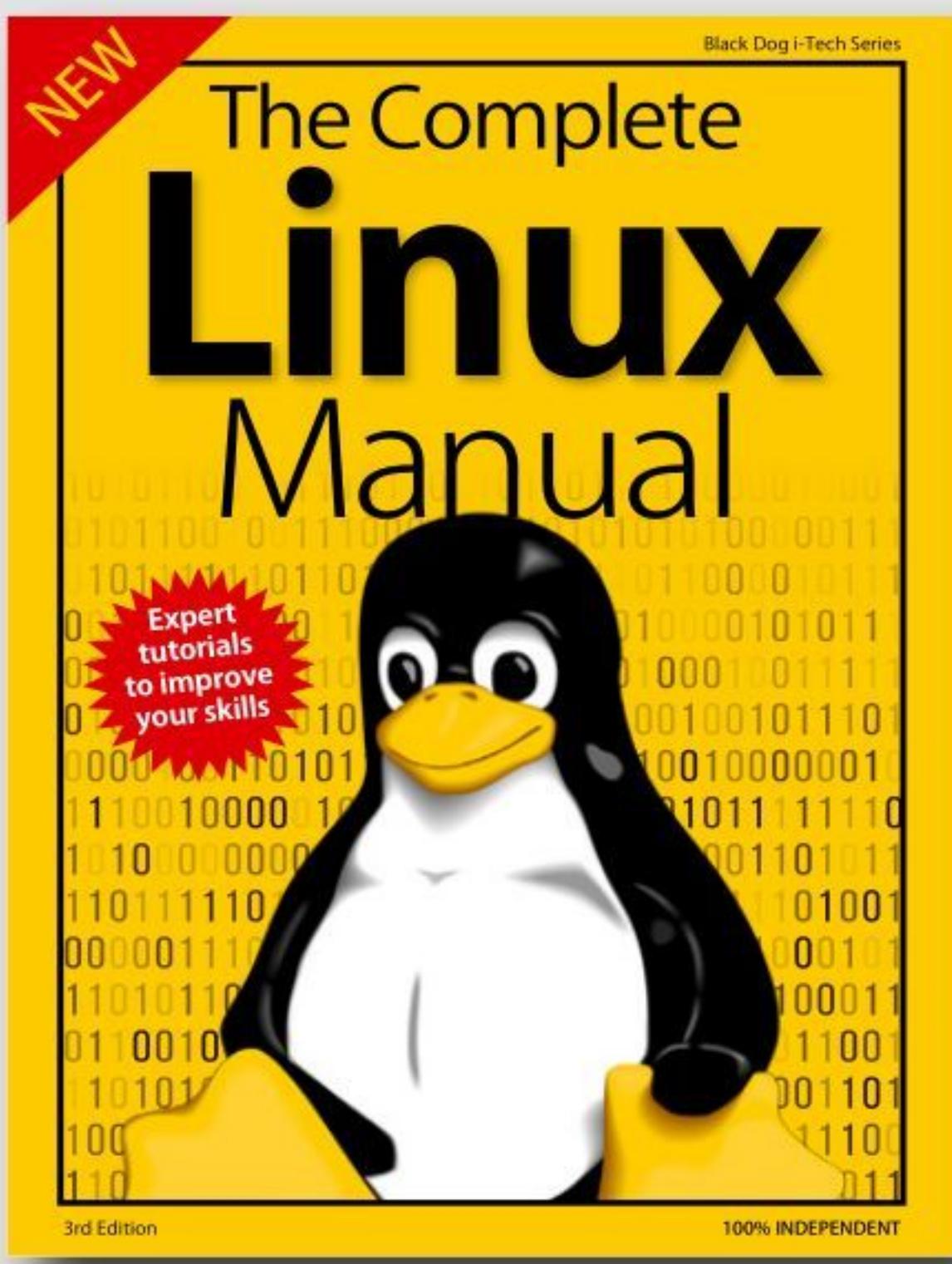
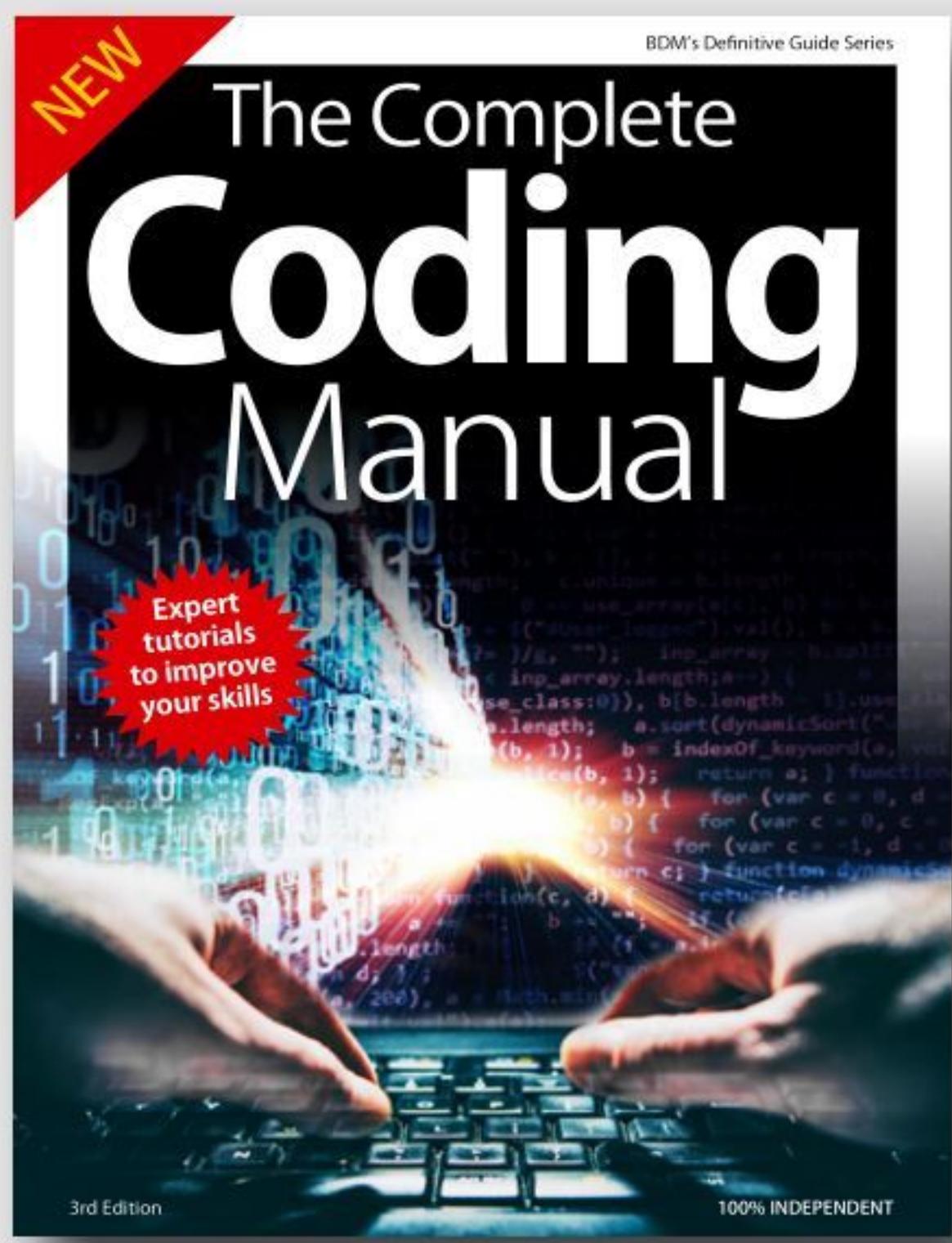
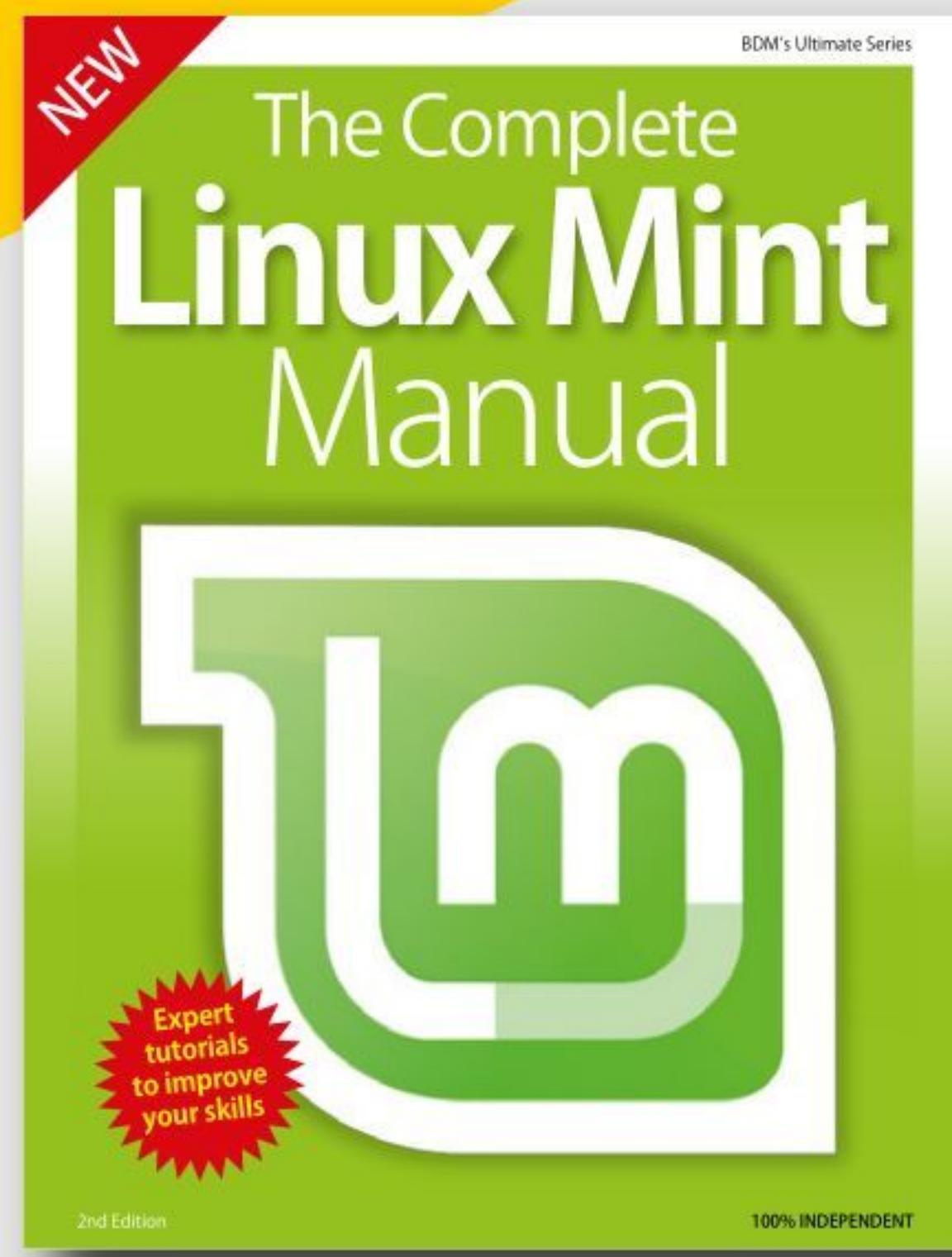
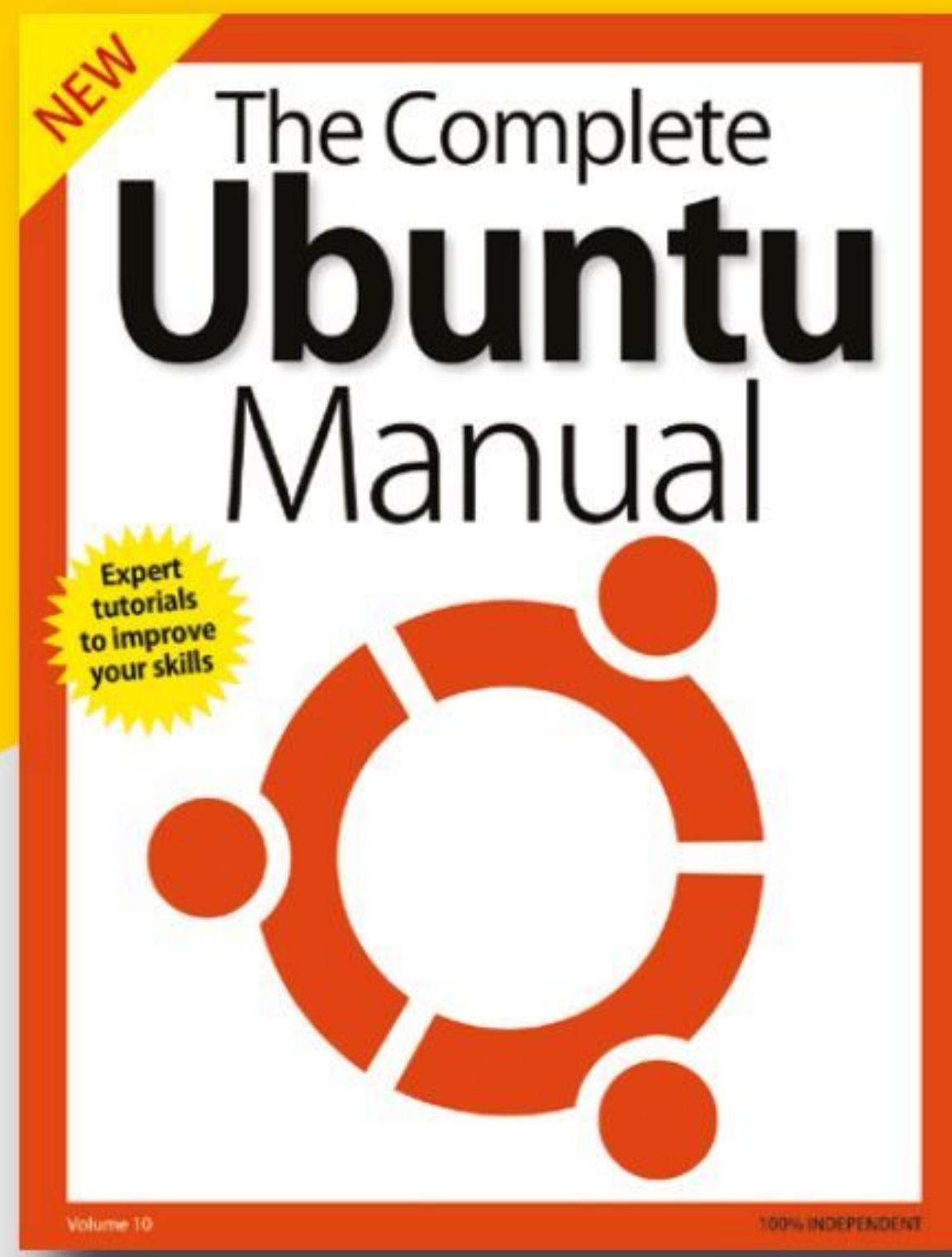
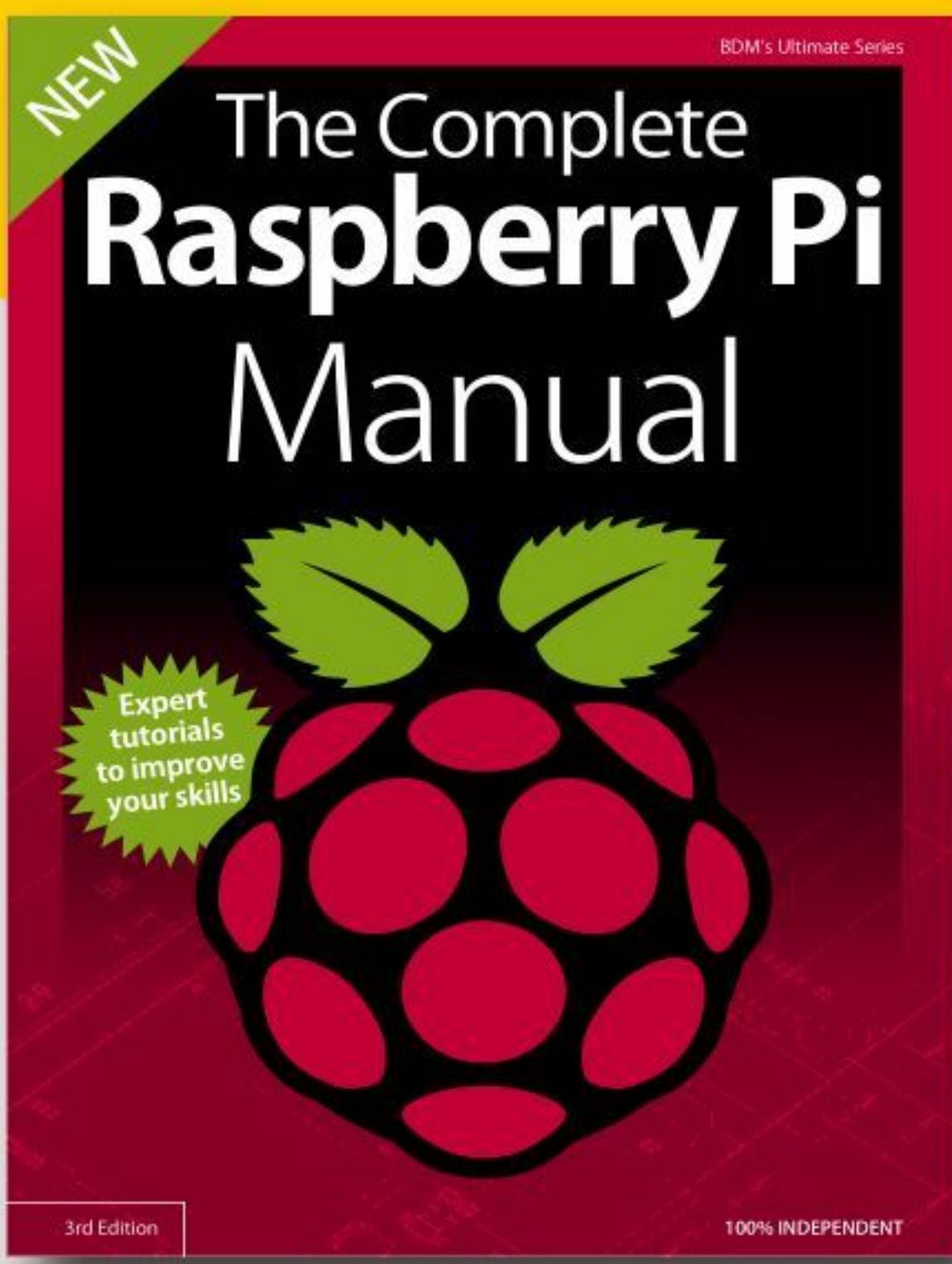
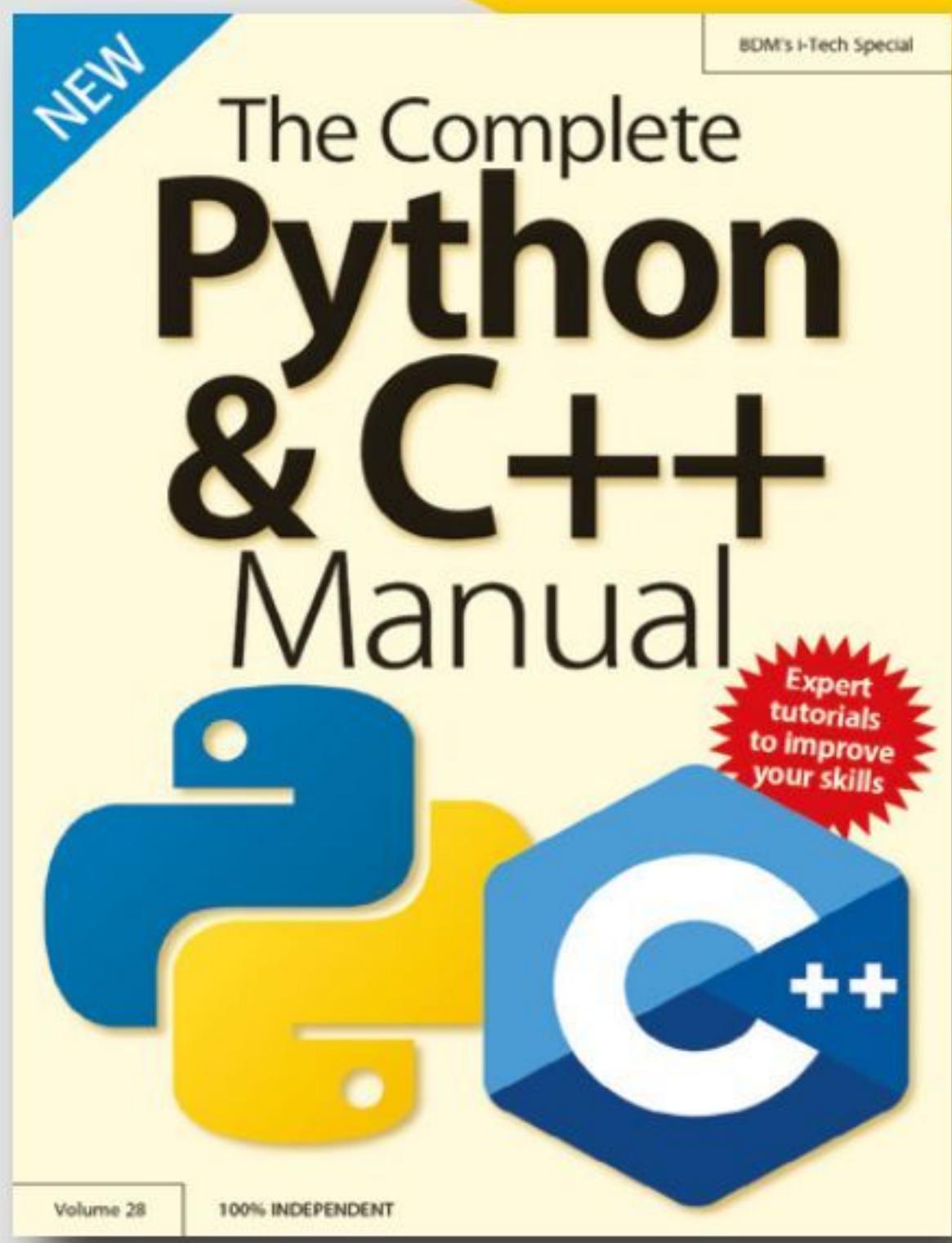
Packed with
top tips and
tutorials

Learn the
latest version
of Python

Featuring Linux
Bash scripts
and code

Start learning
C++, for better
coding

Discover more of our complete manuals on **Readly** today...





The Essential Coding Manual

Your Essential Guide to Coding...

Code is the lifeblood that courses through the veins of our modern digital world. Coding is the power behind nearly everything you interact with, whether it's your TV, washing machine, or even your microwave. It's the driving force behind the Internet and it's used by NASA and other space agencies; CERN scientists utilise millions of lines of code to make sense of the voluminous data from the Large Hadron Collider, and engineers the world over need code to help design and build everything from two-bed houses to earthquake-proof skyscrapers.

The Essential Coding Manual is your gateway to how code works, how you can apply it to your own projects and how you can expand your knowledge and skills. Within these pages are tutorials, hints, tips, and ideas that cover Python, C++ and Linux Bash Scripting and which will help build a solid foundation that you can use to better understand our digital world, or apply to your future job prospects.

Let's start your coding adventure, where the only limits are your own imagination.

60 Python programs
21,500+ lines of code

Master Python
with the help of our fantastic
Code Portal, featuring code for
games, tools and more.

Visit: [https://bdmpublications.com/
code-portal](https://bdmpublications.com/code-portal), and log in
to get access!



Contents

The Essential
</>
Coding Manual

6 The World of Code

- 8 Being a Programmer
- 10 A Brief History of Coding
- 12 Choosing a Programming Language
- 14 Creating a Coding Platform
- 16 Using Virtual Machines
- 18 Equipment You Will Need



Take your first tentative steps into the big wide world of coding, here's what you need.

20 Coding with Python

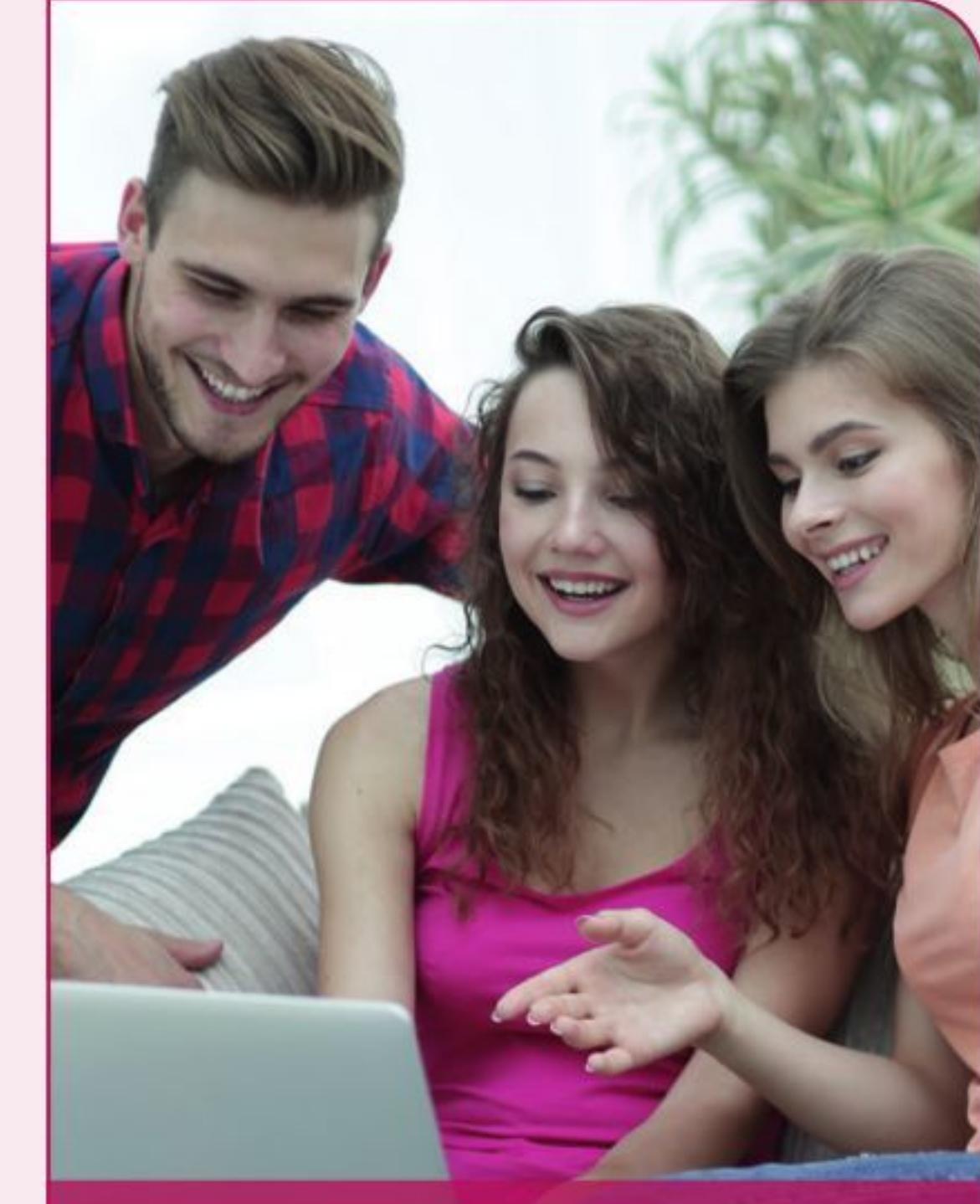
- 22 Why Python?
- 24 What can you do with Python?
- 26 Python in Numbers
- 28 How to Set Up Python in Windows
- 30 How to Set Up Python in Linux
- 32 Python on the Pi
- 34 Getting to Know Python



Learn how to install Python, the most popular programming language in the world.

36 Getting Started with Python

- 38 Starting Python for the First Time
- 40 Your First Code
- 42 Saving and Executing Your Code
- 44 Executing Code from the Command Line
- 46 Numbers and Expressions
- 48 Using Comments
- 50 Working with Variables
- 52 User Input
- 54 Creating Functions
- 56 Conditions and Loops
- 58 Python Modules
- 60 Python Errors
- 62 Combining What You Know So Far



Write your first piece of code, save it and then see how it runs. Welcome to Python!

64 Working with Data

- 66 Lists
- 68 Tuples
- 70 Dictionaries
- 72 Splitting and Joining Strings
- 74 Formatting Strings
- 76 Date and Time
- 78 Opening Files
- 80 Writing to Files
- 82 Exceptions
- 84 Python Graphics
- 86 Combining What You Know So Far



How to: create, store, use and manipulate data. Learn functions and even create graphics.

88 Using Modules

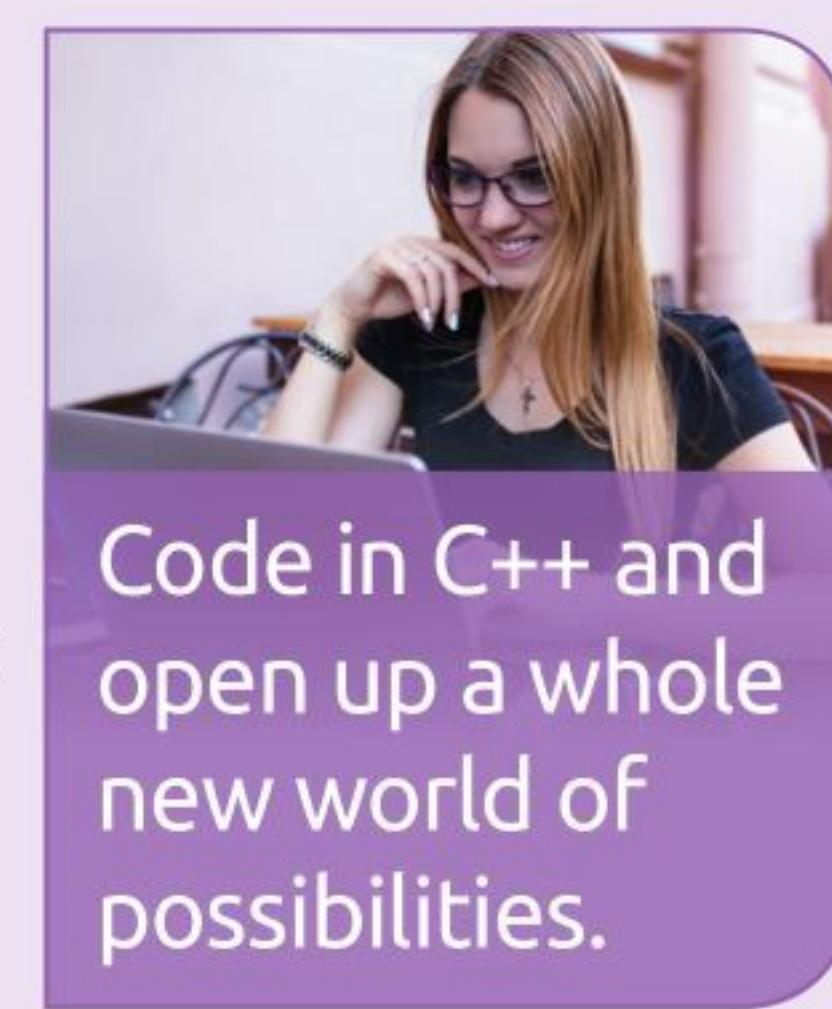
- 90** Calendar Module
- 92** OS Module
- 94** Using the Math Module
- 96** Random Module
- 98** Tkinter Module
- 100** Pygame Module
- 104** Basic Animation
- 106** Create Your Own Modules



Use modules in Python to forge fantastic programs.

108 Coding with C++

- 110** Why C++?
- 112** Equipment You Will Need
- 114** How to Set Up C++ in Windows
- 116** How to Set Up C++ on a Mac
- 118** How to Set Up C++ in Linux
- 120** Other C++ IDEs to Install

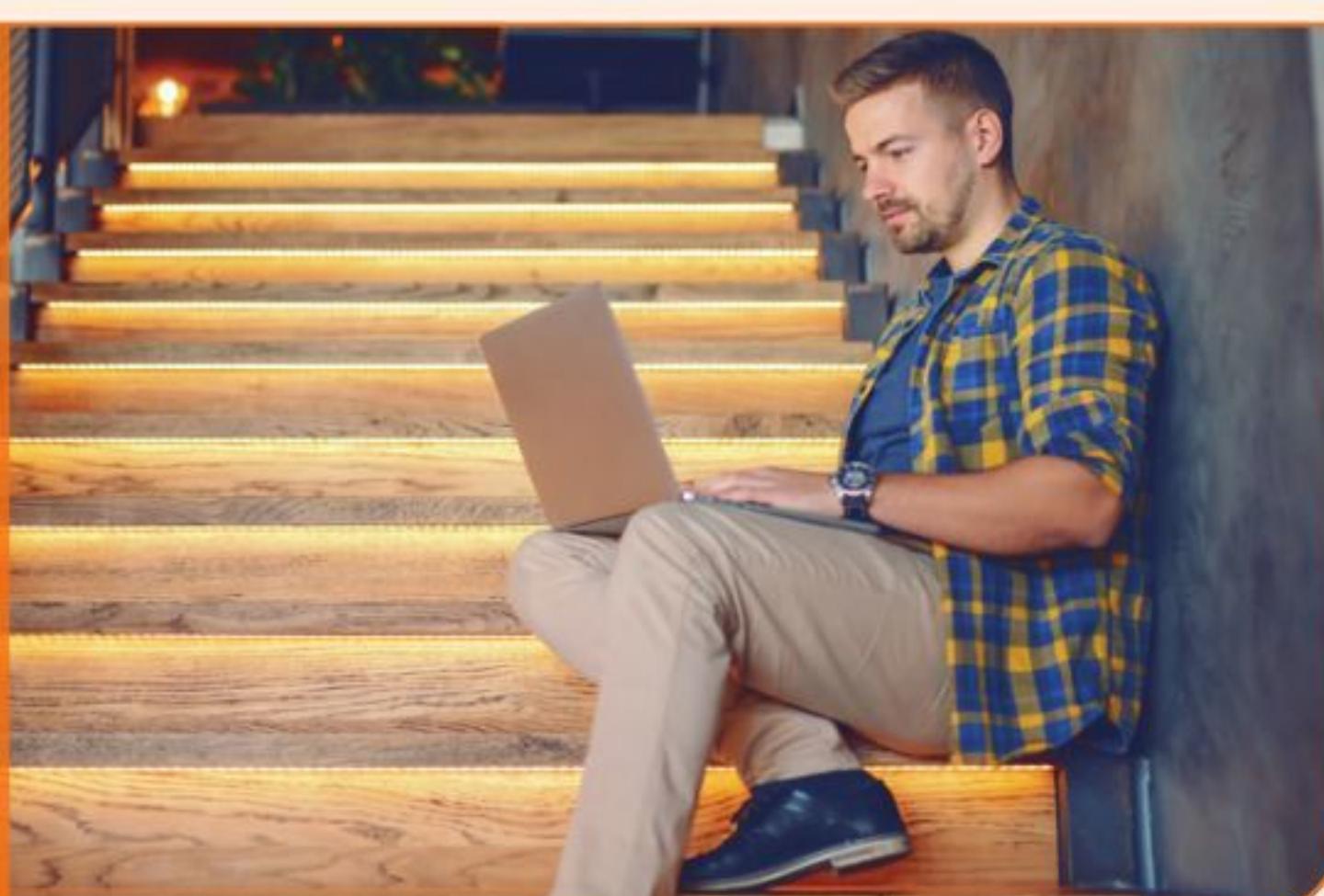


Code in C++ and open up a whole new world of possibilities.

122 C++ Fundamentals

- | | |
|---------------------------------------|-----------------------|
| 124 Your First C++ Program | 132 Variables |
| 126 Structure of a C++ Program | 134 Data Types |
| 128 Compile and Execute | 136 Strings |
| 130 Using Comments | 138 C++ Maths |

Understand the basic structure of C++, how it all comes together, and how to execute the code.

**140 C++ Input/Output**

- | | |
|-------------------------------|-------------------------------|
| 142 User Interaction | 146 Defining Constants |
| 144 Character Literals | 148 File Input/Output |



Add that personal touch, with improved user interaction in your C++ code.

150 Loops and Decision Making

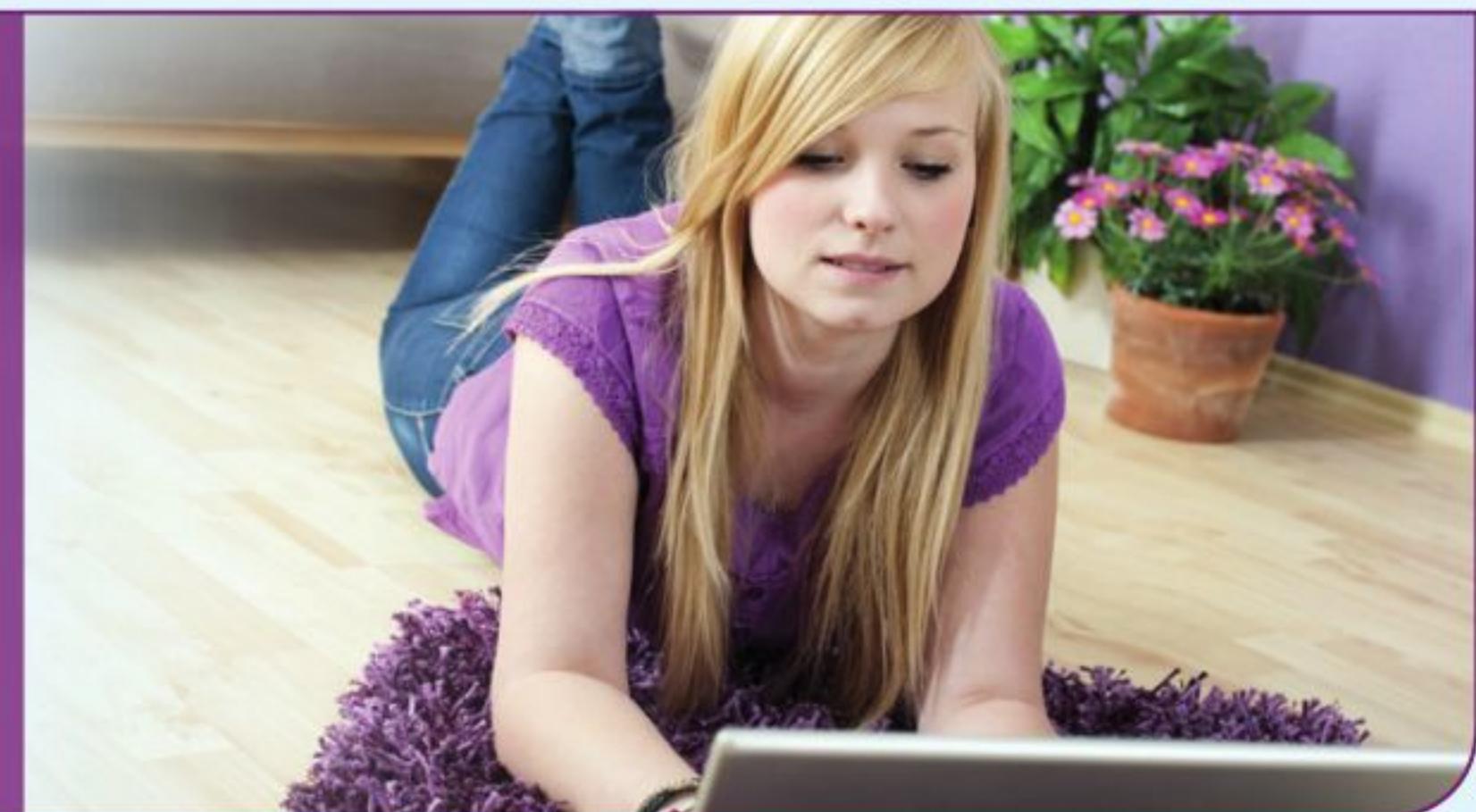
- 152** While Loop
- 154** For Loop
- 156** Do... While Loop
- 158** If Statement
- 160** If... Else Statement

Loops give the code a beginning, middle, and end.

**162 Coding on Linux**

- 164** Getting Ready to Code in Linux
- 166** Creating Bash Scripts - Part 1
- 168** Creating Bash Scripts - Part 2
- 170** Creating Bash Scripts - Part 3
- 172** Creating Bash Scripts - Part 4
- 174** Creating Bash Scripts - Part 5

Get to grips with coding on Linux operating systems.

**176 Coding Projects & Tips**

- | | |
|-------------------------------------------|-------------------------------------|
| 178 Creating a Loading Screen | 188 Common Coding Mistakes |
| 180 Text Animations | 190 Beginner Python Mistakes |
| 182 Tracking the ISS with Python | 192 Beginner C++ Mistakes |
| 186 Using Text Files for Animation | |



Project ideas and tutorials; track the ISS, or create a loading screen? It's all in here.

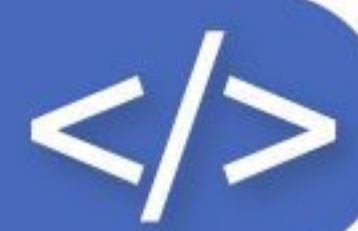


The World of Code



Where to start learning how to code?
Remarkably, this is the most difficult step. There are plenty of programming languages out there, and a seemingly unlimited number of tools to help you achieve what you want; but where do you begin?

-
- 8** Being a Programmer
- 10** A Brief History of Coding
- 12** Choosing a Programming Language
- 14** Creating a Coding Platform
- 16** Using Virtual Machines
- 18** Equipment You Will Need



Being a Programmer

Programmer, developer, coder, they're all titles for the same occupation, someone who creates code. What they're creating the code for can be anything from a video game to a critical element on-board the International Space Station. How do you become a programmer though?





Times have changed since programming in the '80s, but the core values still remain.

"It's up to you how far to take your coding adventure!"

```

1 #include<stdio.h>
2 #include<dos.h>
3 #include<stdlib.h>
4 #include<conio.h>
5 void setup()
6 {
7     textcolor(BLACK);
8     textbackground(15);
9     clrscr();
10    window(10,2,70,3);
11    cprintf("Press X to Exit, Press Space to Jump");
12    window(62,2,80,3);
13    cprintf("SCORE : ");
14    window(1,25,80,25);
15    for(int x=0;x<79;x++)
16        cprintf("\n");
17    textcolor(0);
18 }
19
20 int t,speed=40;
21 void ds(int jump=0)
22 {
23     static int a=1;
24
25     if(jump==0)
26         t=0;
27     else if(jump==2)
28         t--;
29     else t++;
30     window(2,15-t,18,25);
31     cprintf("          ");
32     cprintf("  МЛПППППМ");
33     cprintf("  ЛННННННН");
34     cprintf("  Л     МННННННН");
35     cprintf("  ЛНМ   МННННННММ ");
36     cprintf("  ННННННННННН   Н ");
37     cprintf("  ННННННННННН   ");
38     if(jump==1 || jump==2){
39         cprintf("  ЛНН  НН  ");
40         cprintf("  ЛМ   ЛМ  ");
41     }else if(a==1)
42     {
43         cprintf("  НННН  ННН  ");
44         cprintf("  ЛМ   ЛМ  ");
45         a=2;
46     }
47     else if(a==2)
48     {
49         cprintf("  ПЛМ  ПЛ  ");
50         cprintf("  ЛМ   ЛМ  ");
51         a=1;
52     }
53     cprintf("          ");
54     delay(speed);
55 }
56 void obj()
57 {

```

Being able to follow a logical pattern and see an end result is one of the most valued skills of a programmer.

MORE THAN CODE

For those of you old enough to remember the '80s, the golden era of home computing, the world of computing was a very different scene to how it is today. 8-bit computers that you could purchase as a whole, as opposed to being in kit form and you having to solder the parts together, were the stuff of dreams; and getting your hands on one was sheer bliss contained within a large plastic box. However, it wasn't so much the new technology that computers then offered, moreover it was the fact that for the first time ever, you could control what was being viewed on the 'television'.

Instead of simply playing one of the thousands of games available at the time, many users decided they wanted to create their own content, their own games; or simply something that could help them with their homework or home finances. The simplicity of the 8-bit home computer meant that creating something from a few lines of BASIC code was achievable and so the first generation of home-bred programmer was born.

From that point on, programming expanded exponentially. It wasn't long before the bedroom coder was a thing of the past and huge teams of designers, coders, artists and musicians were involved in making a single game. This of course led to the programmer becoming more than simply someone who could fashion a sprite on the screen and make it move at the press of a key.

Naturally, time has moved on and with it the technology that we use. However, the fundamentals of programming remain the same; but what exactly does it take to be a programmer?

The single most common trait of any programmer, regardless of what they're doing, is the ability to see a logical pattern. By this we mean someone who can logically follow something from start to finish and envisage the intended outcome. While you may not feel you're such a person, it is possible to train your brain into this way of thinking. Yes, it takes time but once you start to think in this particular way you will be able to construct and follow code.

Second to logic is an understanding of mathematics. You don't have to be at a genius level but you do need to understand the rudiments of maths. Maths is all about being able to solve a problem and code mostly falls under the umbrella of mathematics.

Being able to see the big picture is certainly beneficial for the modern programmer. Undoubtedly, as a programmer, you will be part of a team of other programmers, and more than likely part of an even bigger team of designers, all of whom are creating a final product. While you may only be expected to create a small element of that final product, being able to understand what everyone else is doing will help you create something that's ultimately better than simply being locked in your own coding cubicle.

Finally, there's also a level of creativity needed to be a good programmer. Again though, you don't need to be a creative genius, just have the imagination to be able to see the end product and how the user will interact with it.

There is of course a lot more involved in being a programmer, including learning the actual code itself. However, with time, patience and the determination to learn, anyone can become a programmer. Whether you want to be part of a triple-A video game team or simply create an automated routine to make your computing life easier, it's up to you how far to take your coding adventure!