Lovereading4kids.co.uk is a book website created for parents and children to make choosing books easy and fun

extracts from

Lift-the-Flap Computers and Coding

Written by

Rosie Dickins

Illustrated by

Shaw Nielsen

Published by

Usborne Publishing Ltd

Please print off and read at your leisure.

Lovereading4kids.co.uk

What's inside?

All computers – from tiny tablets to huge supercomputers – are built up out of similar bits and pieces, known as **components**. Many of those components depend, in turn, on **chips** – little pieces of electronics which are the 'brains' of the computer.

How do chips work?

Each chip is crammed with microscopic circuits full of switches. When you turn on a computer, electricity flows into its chips. The circuits form a path for the electricity. The switches are like gates, deciding whether the electricity gets through.

Chip

On or off?

The switches in chips are always either on or off. On-off switches might sound simple, but they allow the computer to do lots of things...

On lights up a dot and off leaves it dark...

On can mean 'yes' and off can mean 'no'...

On stands for '1' and off stands for '0'...

Screen

The screen allows the computer to show its results – known as the **output**.

Many tablet computers have touchscreens, so you can use the screen to give instructions, too.

Power switch

Speaker

Keyboard

When you turn on the power, the computer spends a few minutes doing checks and starting up software. This is known as **booting up**.

The keyboard and trackpad (or mouse, if you use one) allow you to give the computer information and instructions – known as the **input**.

Trackpad

Open the case to see the components.

USB (Universal Serial Bus) sockets allow you to plug in extras such as a printer, portable memory stick or camera.

An **optical disk drive** allows the computer to read information from CDs and DVDs.

USB memory stick

Optical disk

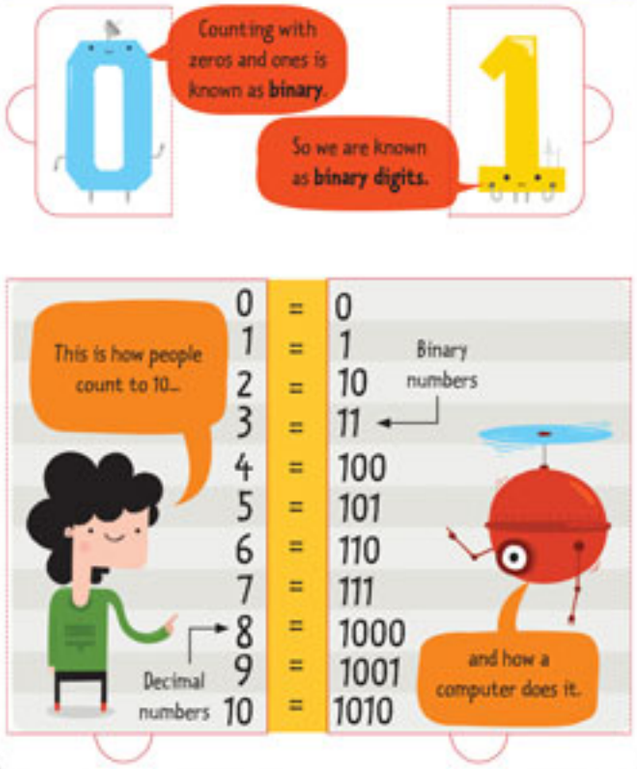
CDs and DVDs are both types of **optical disks**. They are used to store information.

Wireless mouse

How computers think

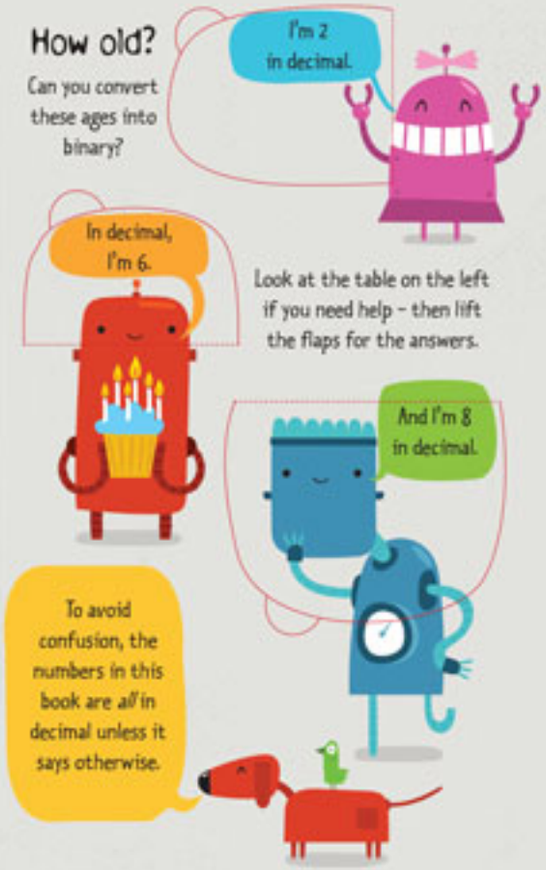
All the information inside computers – even words and pictures – is stored as numbers. And because computers depend on on-off switches, those numbers use just two digits: 0 and 1.

How computers count



How old?

Can you convert these ages into binary?

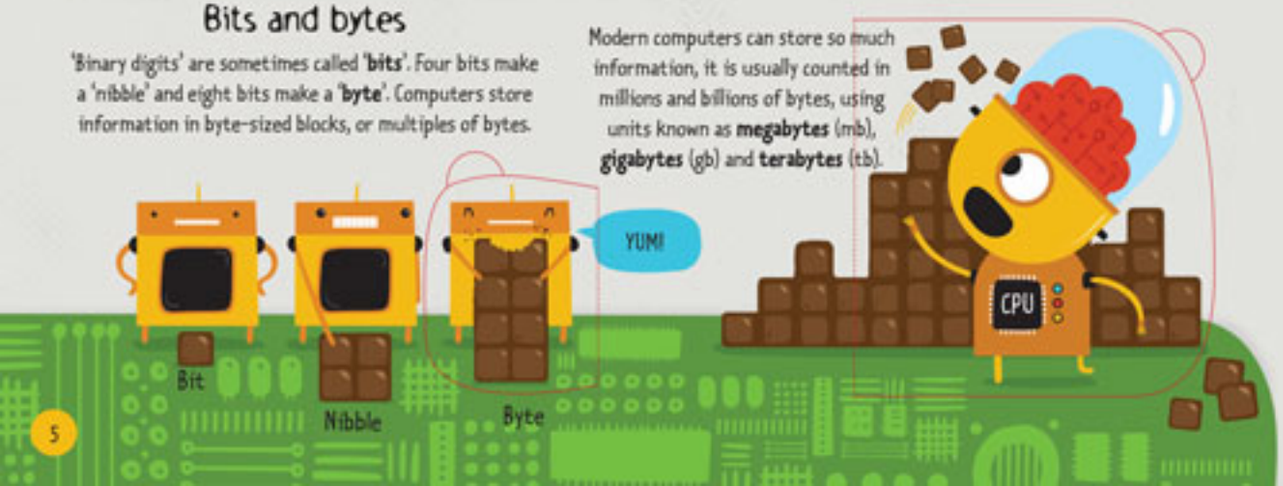


Look at the table on the left if you need help – then lift the flaps for the answers.

Bits and bytes

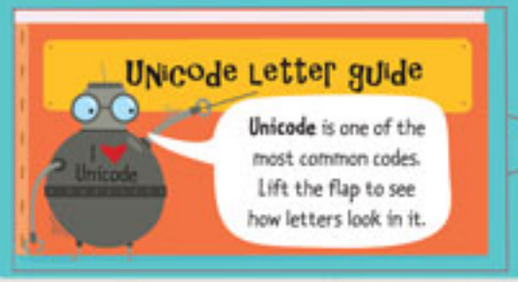
'Binary digits' are sometimes called '**bits**'. Four bits make a 'nibble' and eight bits make a '**byte**'. Computers store information in byte-sized blocks, or multiples of bytes.

Modern computers can store so much information, it is usually counted in millions and billions of bytes, using units known as **megabytes** (mb), **gigabytes** (gb) and **terabytes** (tb).



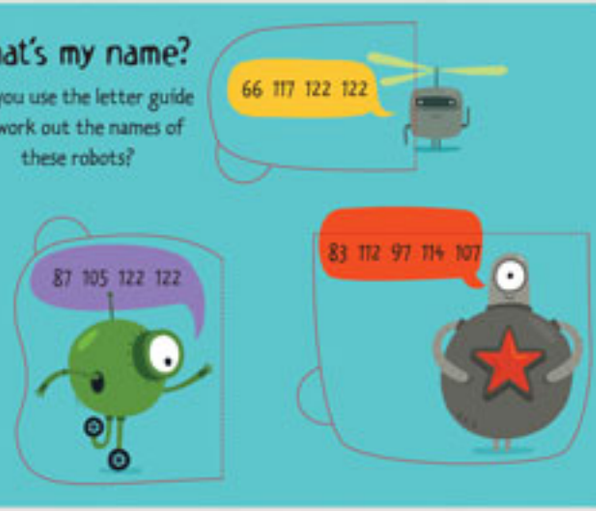
Storing text

To store text, computers generally use a standard code to turn the words into numbers. Following the code, each letter or character (including punctuation and spaces) is given its own number.



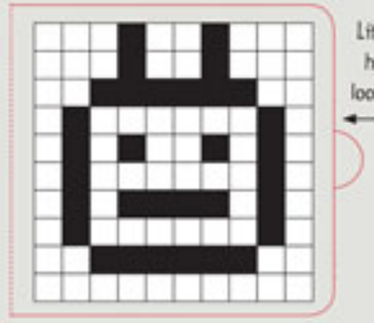
What's my name?

Can you use the letter guide to work out the names of these robots?



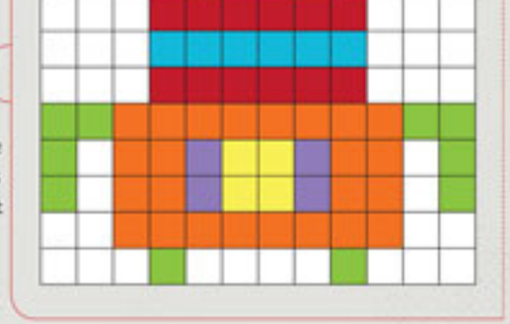
Storing pictures

To store pictures, computers turn them into a grid of tiny dots known as **pixels**. Each line of pixels is then turned into numbers, depending on colour and position.



Lift the flap to see how this drawing looks to a computer.

The more colours there are, the more numbers the computer uses. Lift the flap to see how.



Find the rocket

Can you use the palette to work out which rocket belongs to which alien?

Colour palette

1 2 3 4 5 6

Talk my language

Although computers 'think' in numbers, it's not very easy to write code that way. Instead, most code is written as words, which the computer then translates.

You can find out more on page 9.

Creating better code

Good code isn't just about getting the computer to do things. It's about doing them quickly and efficiently, so you don't have to spend hours coding or waiting for results.

Getting results

In coding, there are always several ways to get the same result. Each way will have good and bad points.

A very simple program may take a long time to run if it involves lots of repeats or extra steps. Adding shortcuts and clever techniques can speed things up, but makes the code itself slower to write.

It's a bit like solving a maze...

I'll follow one wall until I find the middle.

I'll map every possible path.

A common shortcut is not bothering to work out exact answers – just getting close enough.

"WHAT'S 355 ÷ 113?"

That's a long division...

But I only need a rough answer.

Getting the best results from a computer, in the best time, is known as **optimisation**.

Time-saving routines

Another time-saving trick is to give any useful piece of code a name and save it. Then you can ask the computer to look it up and use it again later. A named piece of code is known as a **routine**.

I've got a great game idea... how do I write it?

Most languages come with **libraries** full of ready-made routines, which you can use in your own programs to save yourself time.

Sharing code

Sharing your code, and seeing how other people write theirs, is a great way to improve your skills. You could join a code club or look online. Go to **www.usborne.com/quicklinks** to find out more.

Dazzling data

Another major issue is managing information – also known as **data**. Computers are constantly using data to work things out. So the data must be organized and labelled, to help the computer find what it needs. This means creating variables and lists...

I can't find anything in this mess!

Unorganized data is like a library with the books all in a muddle.

Valuable variables

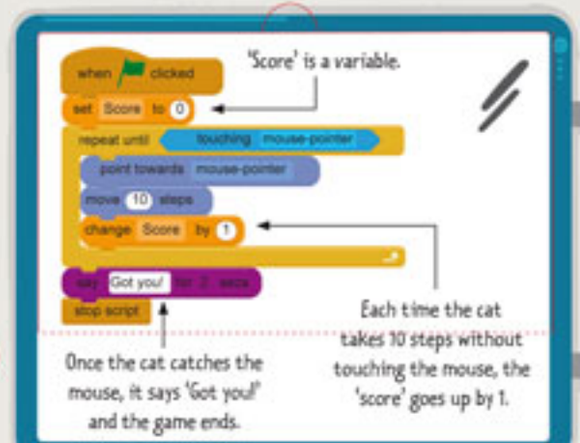
A **variable** is a piece of information that has been given a name – such as a 'score' in a game. The name makes it easier to keep track of that information – especially if its value changes.

A variable is like a box with a label...

MY STUFF

You can use variables to adapt the program from page 10 into a simple Cat and Mouse game, like this...

How to play Cat and Mouse



Lengthy lists

If there are lots of variables, you will need to organize them. The easiest way is to group them into a **list** – then sort the list into an order.

Here is an unsorted list. Can you find the highest score?

Sparky's Score

Bash's Score

Wizz's Score

Zoom's Score

How many flaps did you need to open?

You can speed things up by sorting the list from lowest to highest...

Now look at the sorted version. How many flaps does it take now?

HIGHER

Wizz's Score

Zoom's Score

Sparky's Score

Bash's Score