

THE
QUESTKIDS[®]



in easy steps
BOOKS FOR KIDS

Coding with the **micro:bit** Create Cool Programming Projects



Dan Aldred

Code a pocket-sized computer to interact with the world around you!



Tiffany Tech

Smuffy

Dan Devices

Chew-Chew

Welcome, budding coders!

Say hi to our heroic adventurers: Dan Devices and Tiffany Tech. Together they are the best of friends, and an amazing team who love to explore and help each other whenever they can. Dan adores devices, Tiff loves tech, and both are on a constant quest to learn and make learning fun. The friends are joined by their pets, Smuffy and Chew-Chew, who help out in their own special way. Together... they are **The QuestKids!**

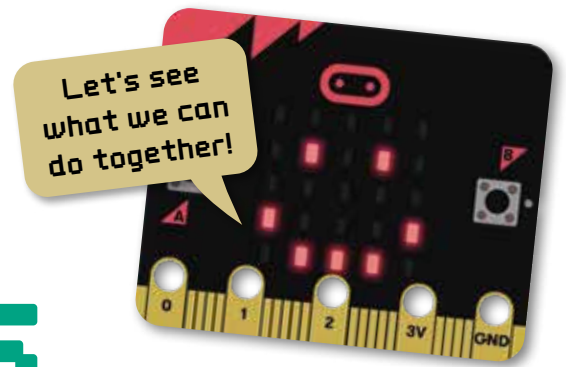
If you're new to coding it can feel a bit daunting, but learning to code with the micro:bit (also known as BBC Micro:bit) is a great way to get started, and to understand how a computer works. With the help of this book, and The QuestKids of course, you'll be up and running in no time!

Tiffany, Dan, Smuffy and Chew-Chew together will take you through the micro:bit's features, and equip you with the skills you need to start coding fast – all **in easy steps.**

All the key functions of the hardware are clearly explained at the start of the book. Then you'll learn key skills before going on to create fun and engaging projects!

About the author: Dan Aldred has more than 20 years of experience in computer science education and teaching. A freelance resource writer and hacker, he has championed the use of computing devices as tools for learning and creativity. Dan is a Raspberry Pi Certified educator, and in 2015 he led the winning team of students for the first Astro Pi competition, whose code is now orbiting Earth on board the International Space Station. Dan currently resides in the UK.

Contents



Welcome to micro:bit coding	4
Skill set 1 Getting started	16
Skill set 2 Basics	26
Skill set 3 Inputs	32
Project 1 > The 99 game	44
Project 2 > Heads or tails	48
Project 3 > Steady hand game	52
Project 4 > Temperature display	56
Project 5 > LED control	62
Project 6 > Trespasser alarm	65
Project 7 > Fortune teller	73
Project 8 > Compass	77
Project 9 > Radio communication	84
Project 10 > Bag alarm	90
Glossary	94
My notes >	96

Welcome to micro:bit coding

To get the best out of this book we encourage you to read it in page order to gain the maximum knowledge and skills. Pages 6-15 will help you get familiar with the micro:bit and the online coding editor you will use to create code. Then you'll move on to learning three skill sets. These skills are needed for you to progress to the projects. Most of all, have fun as you go!

WHAT YOU'LL NEED FOR THE PROJECTS

- For some projects two micro:bits are required
- 2 x battery packs
- 4 x AAA batteries
- 4 x crocodile clips with wires
- An LED
- An elastic or rubber band
- Temperature sensor TMP36
- Foil
- A speaker or headphones
- Your school bag

SKILLS AND PROJECTS

There are 10 interactive projects for you to try. At the start of each one is a panel with the skills you'll learn and the equipment needed. Then there is a simple, visual set of instructions to follow, with some theory to explain how things work. All the code is given to you, so even if you've never programmed before, you'll be able to follow along. Here's a quick breakdown of your book:

SKILL SETS

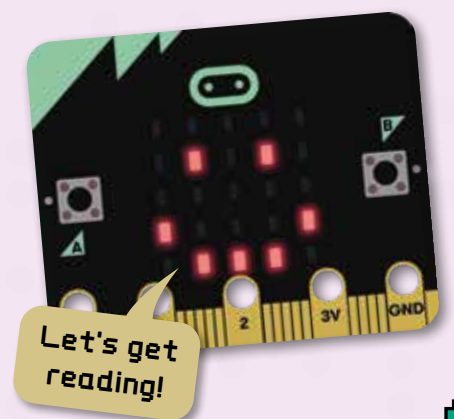
1 > Getting started This gives you a tour of the micro:bit and gets you started with coding your first program.

2 > Basics Here, you'll learn how to display images and text on the micro:bit's LED grid.

3 > Inputs Get to grips with the input systems such as the buttons, accelerometer, and light sensor.

PROJECTS

- 1 > **The 99 game** This project is a game of chance that involves throwing the micro:bit among a group of friends. You'll learn more about coding, like using variables and control flow techniques.
- 2 > **Heads or tails** You'll use the LED grid and accelerometer to create a "coin-toss" generator that detects when you flip the micro:bit, and it displays heads or tails.
- 3 > **Steady hand game** This game challenges you to hold the micro:bit in your hand as long as you can without wobbling. You'll learn more about the micro:bit's abilities and other fundamental coding techniques.
- 4 > **Temperature display** Use the built-in temperature sensor to detect the temperature and scroll it across the LED grid.
- 5 > **LED control** Learn how to wire up an external LED and control it with code. This exercise teaches you about using the micro:bit with other components so you can build even bigger projects.
- 6 > **Trespasser alarm** By connecting a speaker to the micro:bit, you'll create an alarm system that detects when someone has broken into your room, and alerts you by playing a tune.
- 7 > **Fortune teller** This project ramps up the coding complexity, showing you how to program your micro:bit to tell your fortune just like a Magic 8 ball!
- 8 > **Compass** Turn the micro:bit into a real working compass using its built-in magnetometer.
- 9 > **Radio communication** Use two micro:bits to send and receive radio messages.
- 10 > **Bag alarm** Using your radio skills from the previous project, combined with the accelerometer, you'll build a bag alarm with two micro:bits. The first micro:bit stays inside your bag and alerts a second micro:bit in your pocket if someone tries to steal your bag.



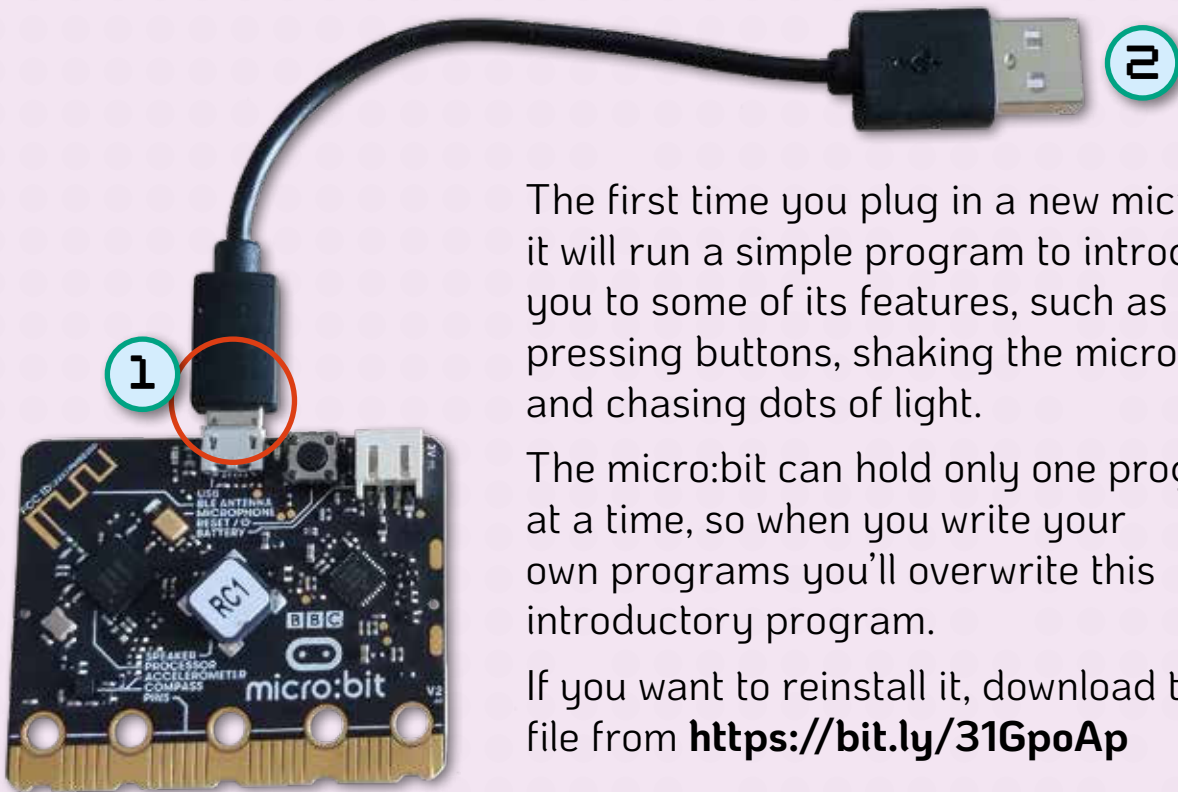
PLUGGING IN YOUR MICRO:BIT

1

Plug the microUSB cable into the port located at the top of your micro:bit.

2

Plug the other end of the cable into a USB port on your computer or laptop. The micro:bit will power up.



The first time you plug in a new micro:bit, it will run a simple program to introduce you to some of its features, such as pressing buttons, shaking the micro:bit, and chasing dots of light.

The micro:bit can hold only one program at a time, so when you write your own programs you'll overwrite this introductory program.

If you want to reinstall it, download the file from <https://bit.ly/31GpoAp>

USING THE MAKECODE EDITOR

Open your web browser (Chrome or Edge is recommended), enter the address <https://makecode.microbit.org>, and click **New Project**. This will take you to the **MakeCode editor** where you can drag and drop **coding blocks** onto the coding area. These blocks snap together to create a program that can be converted into JavaScript or Python code.

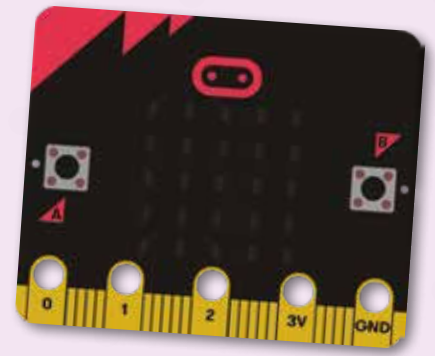
The advantage of using blocks is that you don't have to worry about typos or syntax; the blocks take care of that. You'll create your program code using this editor, which will automatically save your progress in your web browser so you can access it again later. You'll then download your programs so you can transfer them to the micro:bit.

Let's look at a few aspects of the MakeCode editor on page 7.



Running the simulator

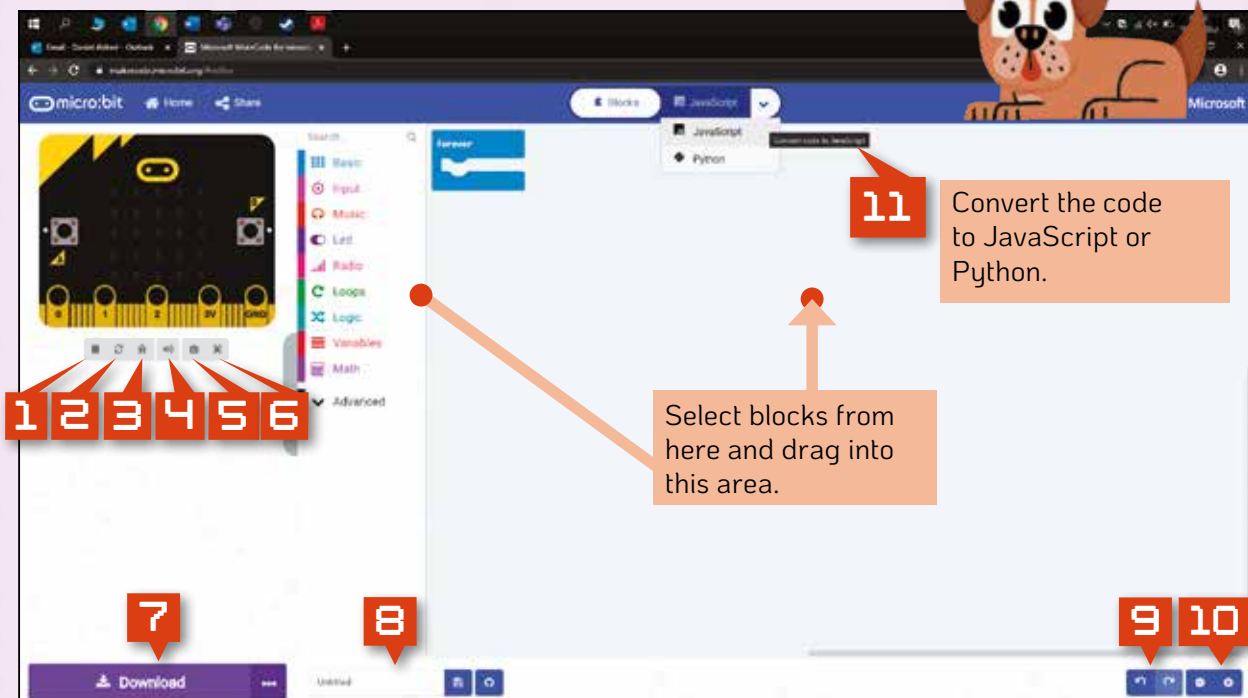
The MakeCode editor has a **micro:bit simulator**, so even if you don't yet have an actual micro:bit you can still write programs and run them. The simulator runs your programs as if it were a real micro:bit – you can even shake it, press its buttons, and attach virtual headphones to it.



The simulator is a great tool for testing your programs work before you download them to the micro:bit.

Navigating the coding interface

Here's a quick overview of the MakeCode editor and what each button does:



- | | |
|--|---|
| <ul style="list-style-type: none"> 1 Stop the micro:bit simulator 2 Reset the simulator 3 Enter debugging mode to run the program in slow motion and fix any errors before downloading it 4 Mute the audio playing through your computer's speakers 5 Run the simulator in full-screen mode | <ul style="list-style-type: none"> 6 Take a screenshot of the micro:bit display 7 Download the program file 8 Rename and save the program file 9 Undo or redo the last block 10 Zoom in or out of the programming window 11 Convert your code to JavaScript or Python |
|--|---|

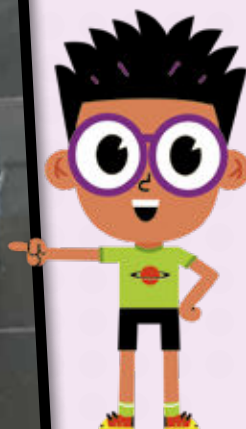
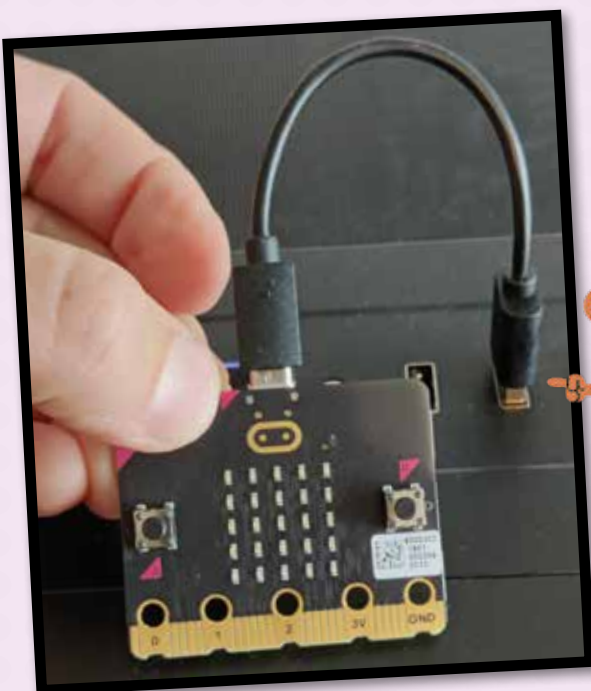
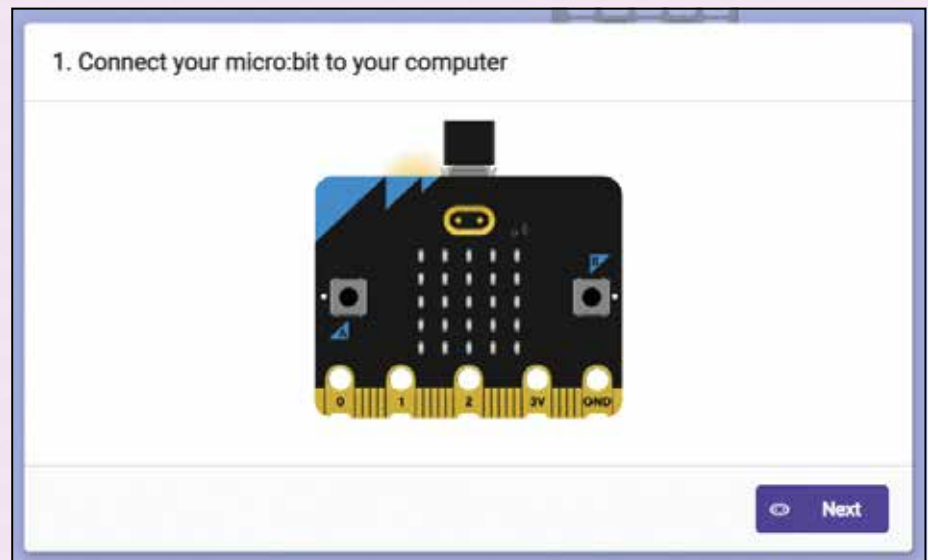
Running your code on the micro:bit

Once you've written and tested a program, you'll want to run it on a real micro:bit. When you click the **Download** button, the MakeCode editor will connect to your micro:bit and attempt to write the program code to your micro:bit.

By default, the file will be given a random name, usually *microbit-Untitled.hex* but it's a good idea to give your project a descriptive name to make it easier to locate the file later. To rename the file, enter a new project name in the **Untitled** text box and then click **Download**.



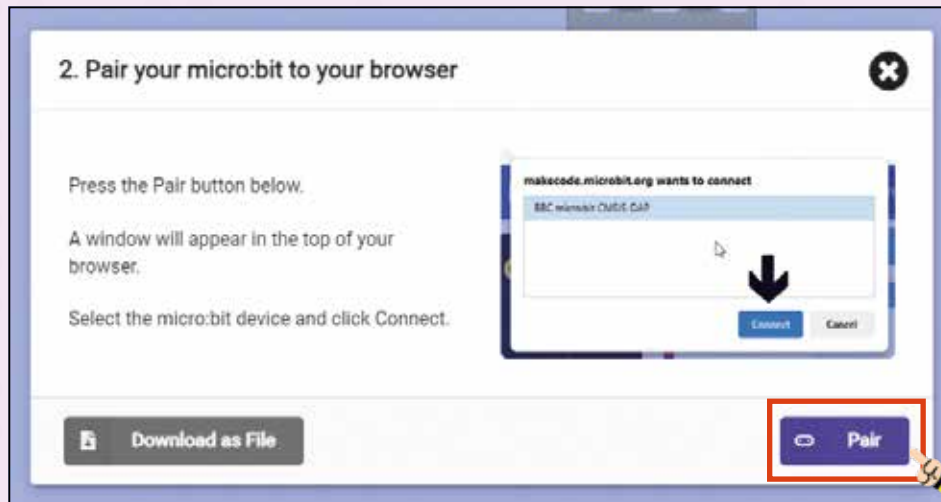
When you click **Download**, you will be asked to connect and pair a micro:bit. This only needs to be done the first time that you connect.



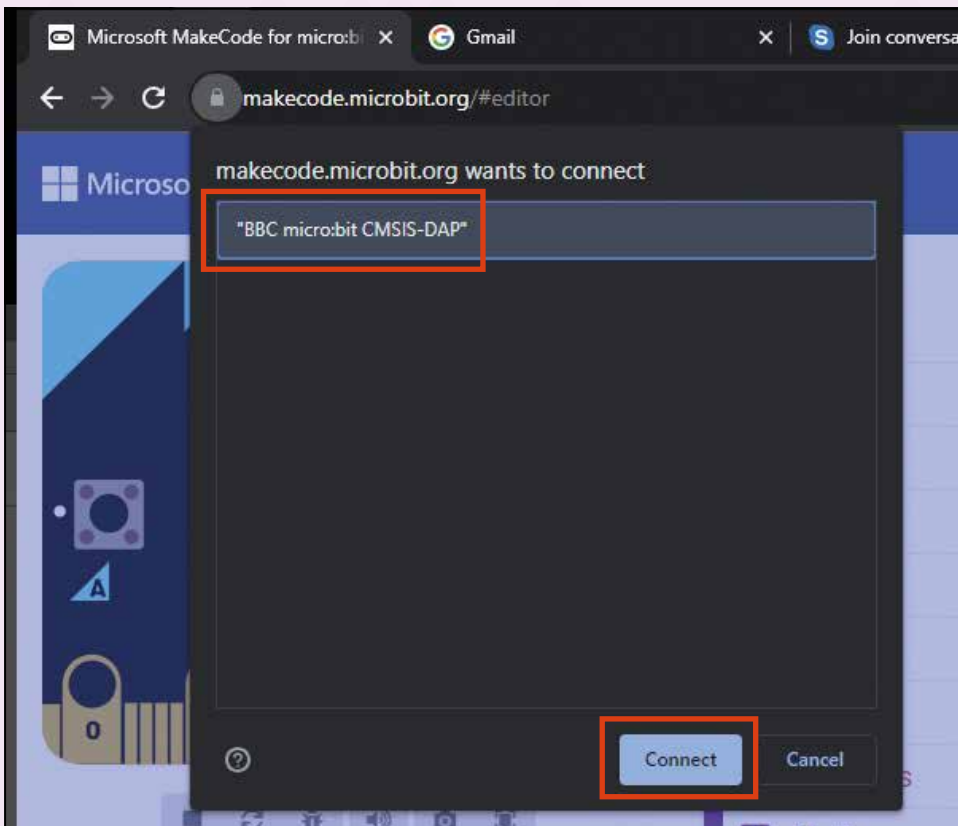
If you haven't already, plug your micro:bit into one of the USB ports on your device. Then click the **Next** button.

micro:bit plugged into a USB port

You will be asked to pair the micro:bit. This simply means to join the micro:bit to your computer so you can transfer code. Press the **Pair** button.



The following window will load. Select BBC micro:bit from the list and then press the **Connect** button.



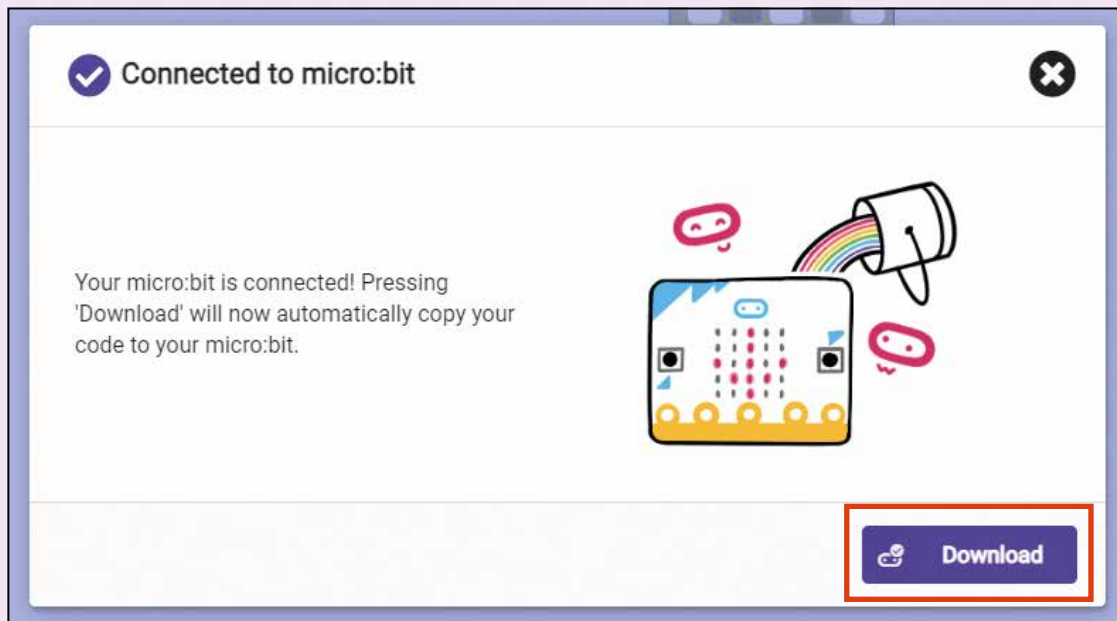
Nearly done pairing.

Now to download your code.

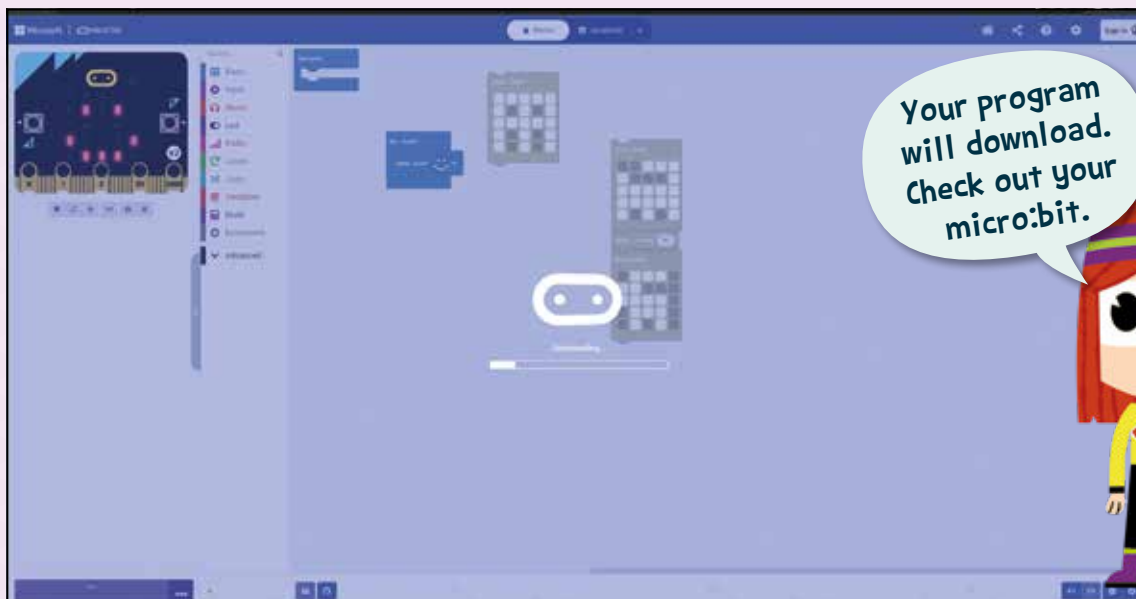


WELCOME TO MICRO:BIT CODING

Once successfully paired you will see the window below. Press the **Download** button to write the program to your micro:bit.



The **Let's code** window will display this downloading page.

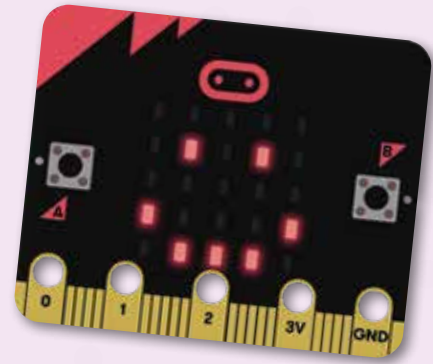


On the back of the micro:bit is a single red LED. When it is on, it means the micro:bit is powered. A flashing LED indicates your program is being written to the micro:bit.

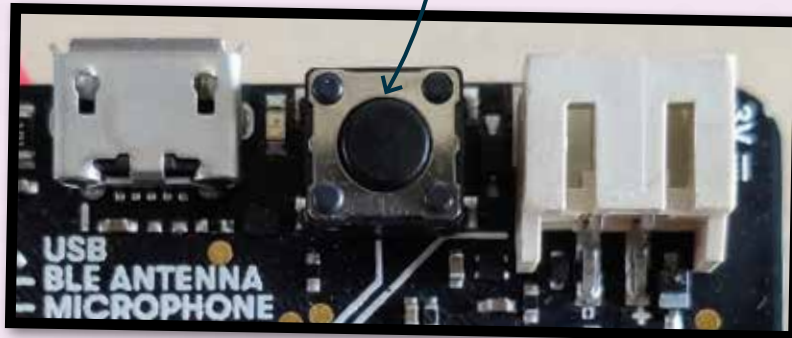


Once the program is downloaded, the LED will stop flashing. Turn the micro:bit over to use your program.

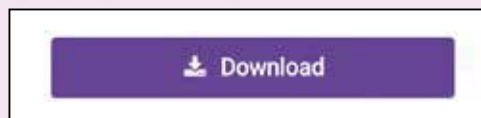
To restart the program, simply press the **Reset** button on the back.



Reset button



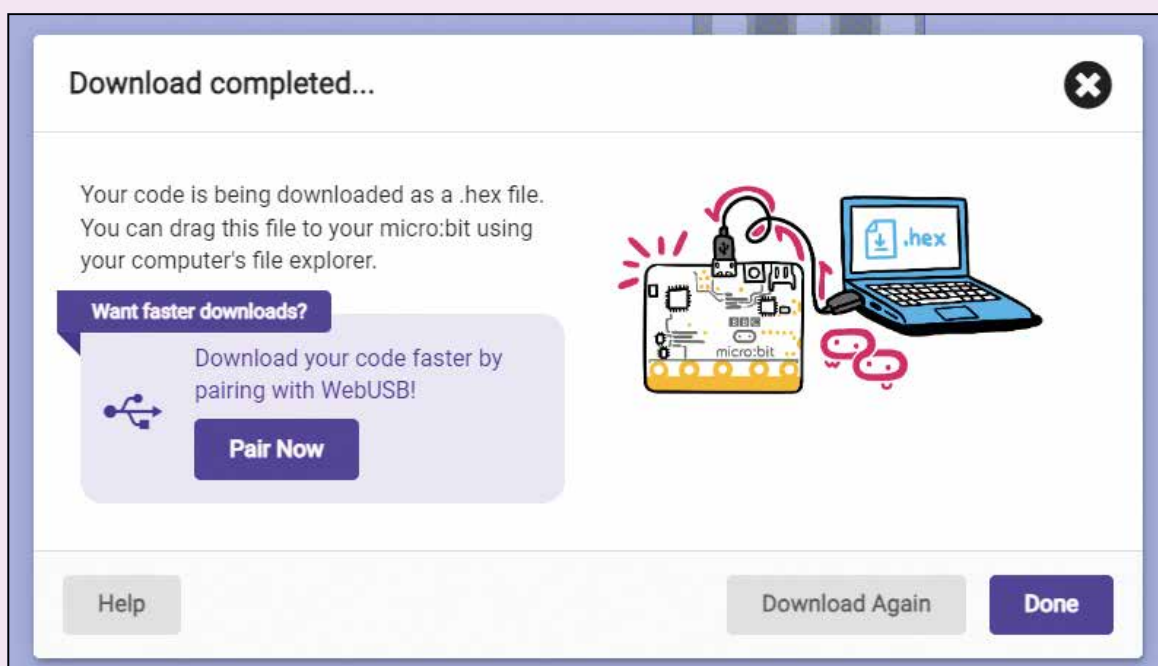
Next time you write your program code to the micro:bit, simply press the **Download** button – there's no need to pair.



Click
Download.

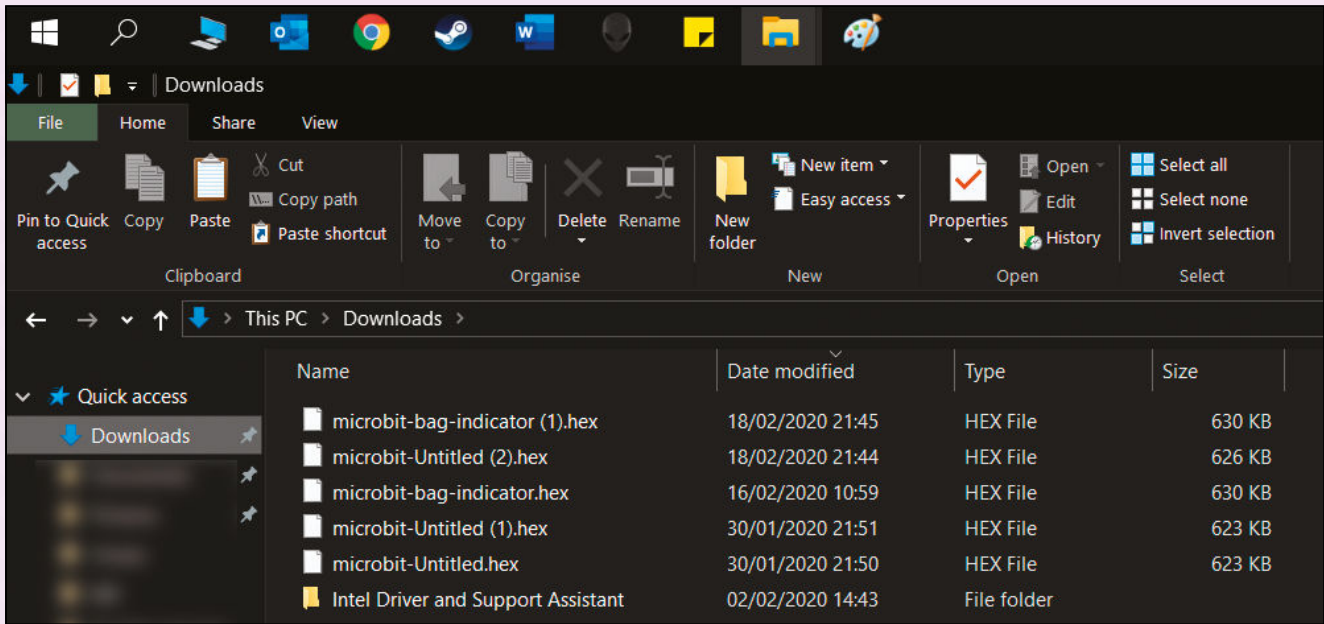


Sometimes you might want to download the program to your computer first. Without the micro:bit plugged in, press **Download**. You will see the screen below and the file will be saved to your *Downloads* folder.

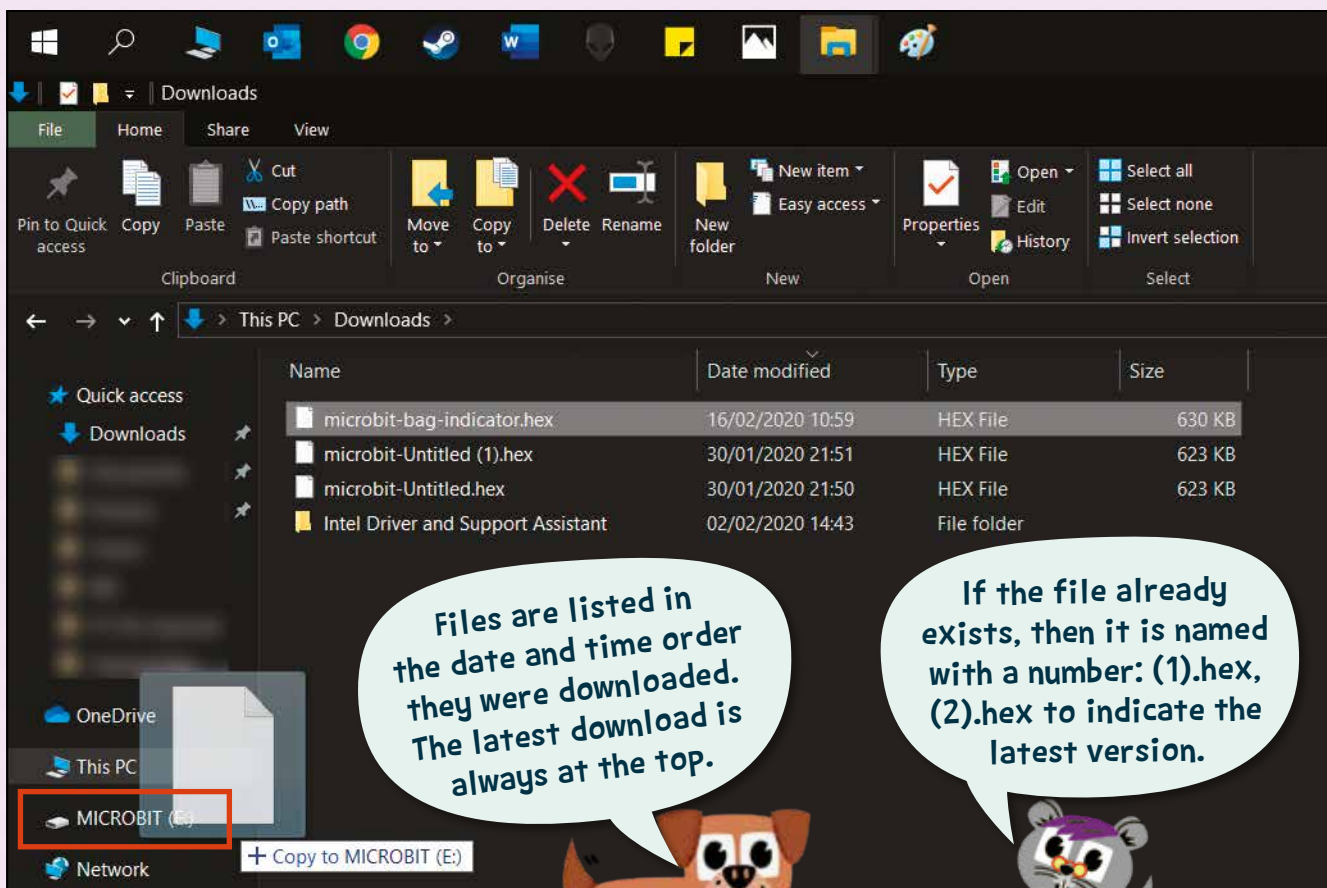


WELCOME TO MICRO:BIT CODING

Open your *Downloads* folder and locate your program file. It should have the name you gave it, and the *.hex* extension.

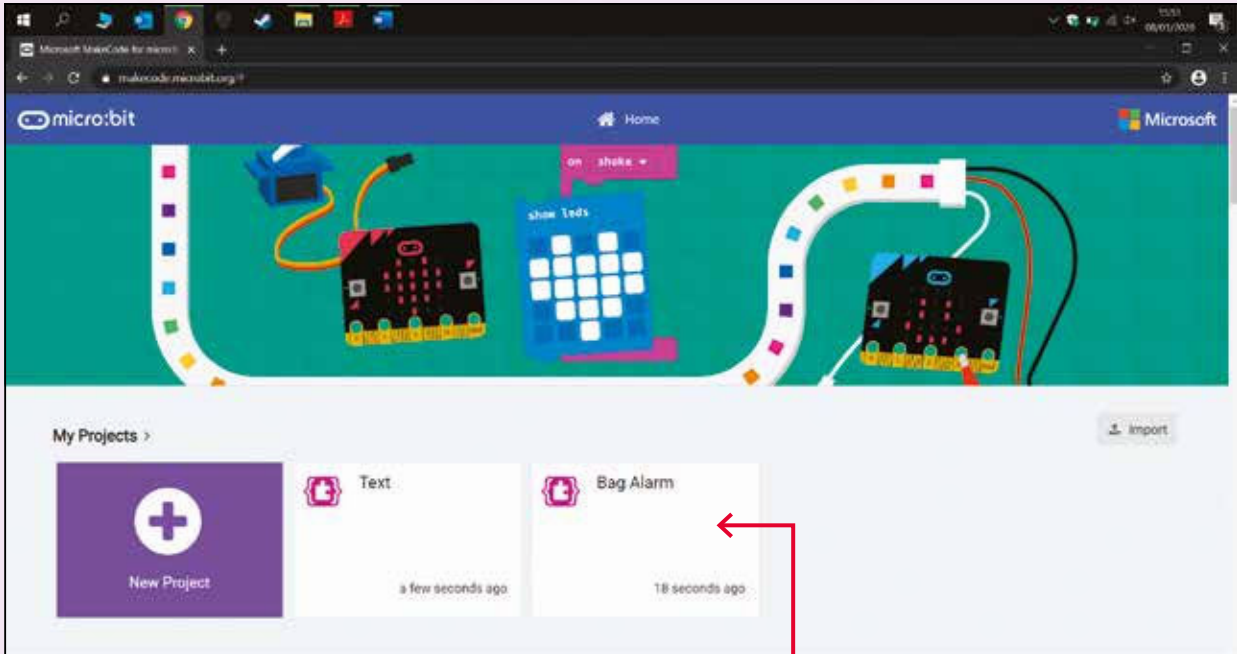


When you plugged in the micro:bit, your computer should have registered it as a drive. That means you should see MICROBIT in the Quick access panel on the left of your screen. Drag the *.hex* file over to the MICROBIT drive (see below) to start downloading the program to the micro:bit. You will see a flashing LED on the back of the micro:bit. This indicates the file is transferring successfully.



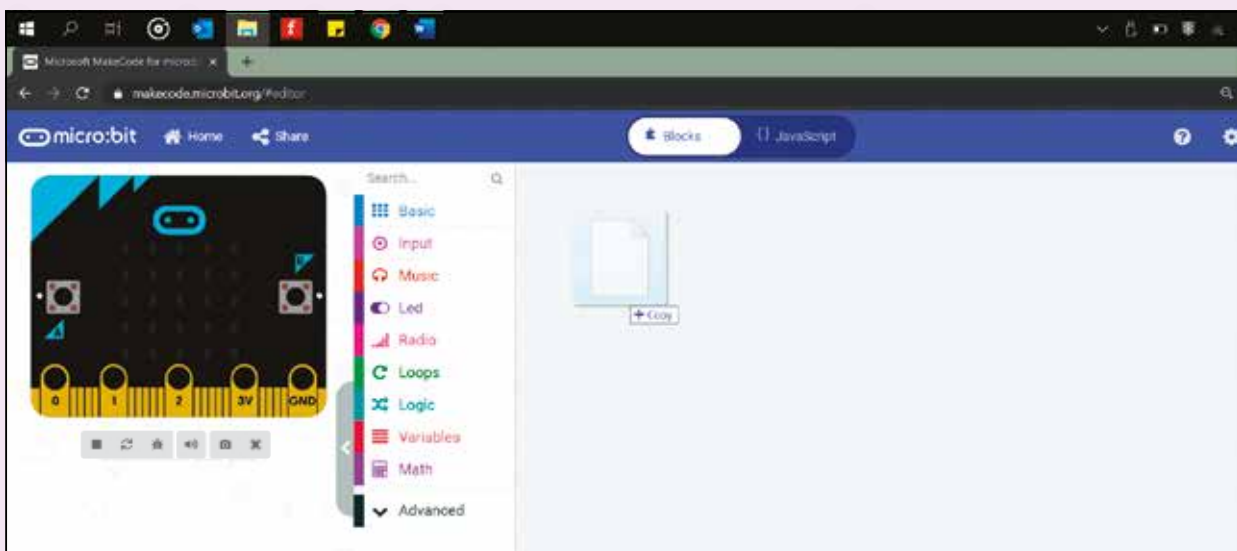
Uploading code

When you return to the MakeCode editor site, it should automatically display a list of projects that you've previously written and saved, as shown below.



Click the name of the program that you want to open.

You can also drag and drop the .hex files from your *Downloads* folder into the coding area to load an old project, as shown below.



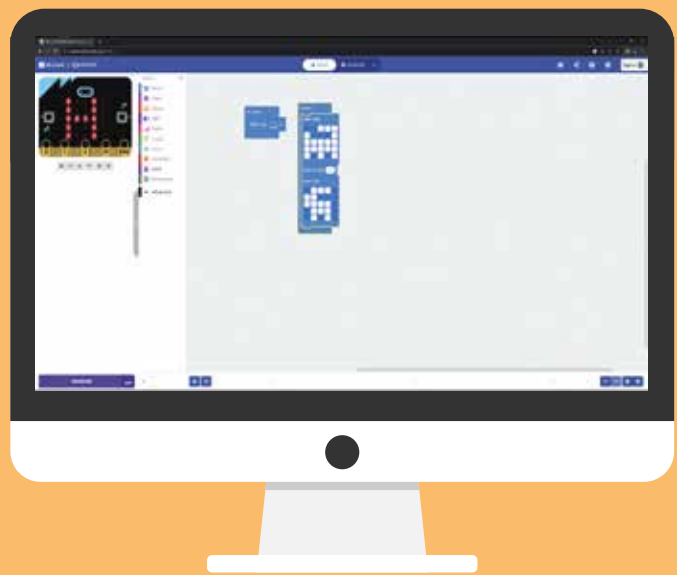
Loading a saved project

NOTE: It is not always possible to load the program file from the micro:bit hardware, so ensure that you save a separate copy on your computer.

SKILL SET 1

Getting started

It's time to get familiar with the micro:bit features and hardware. You'll also learn how to access the MakeCode editor to write your first program.



IN THIS SKILL SET

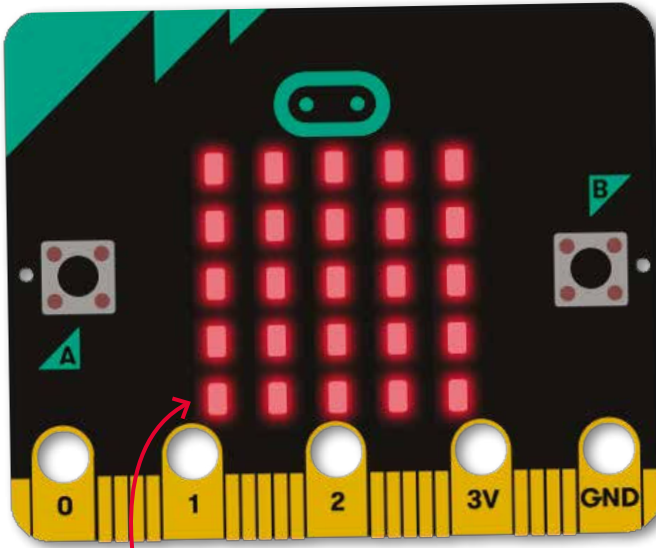
- 1 Meet the micro:bit
- 2 Get connected
- 3 Access the coding interface
- 4 Explore the MakeCode editor
- 5 Start a program
- 6 Show a string
- 7 Use the micro:bit simulator
- 8 Write the program to your micro:bit
- 9 View the JavaScript or Python code

Are you ready to take a closer look at the micro:bit?



1 Meet the micro:bit

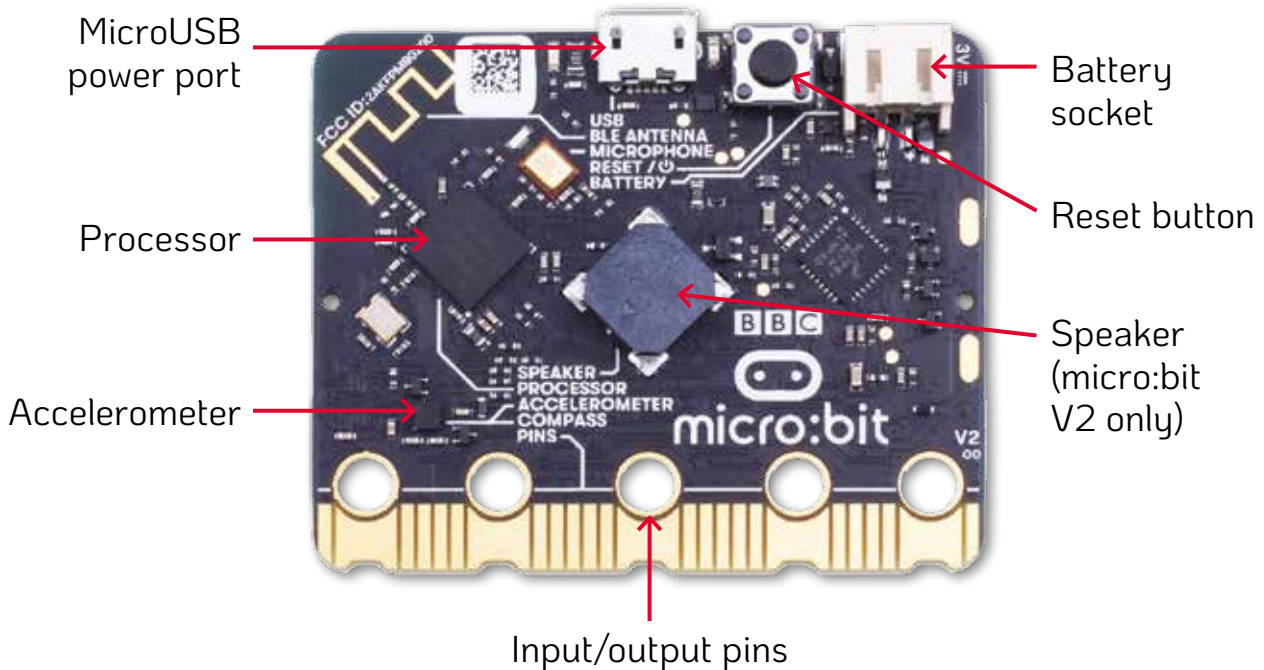
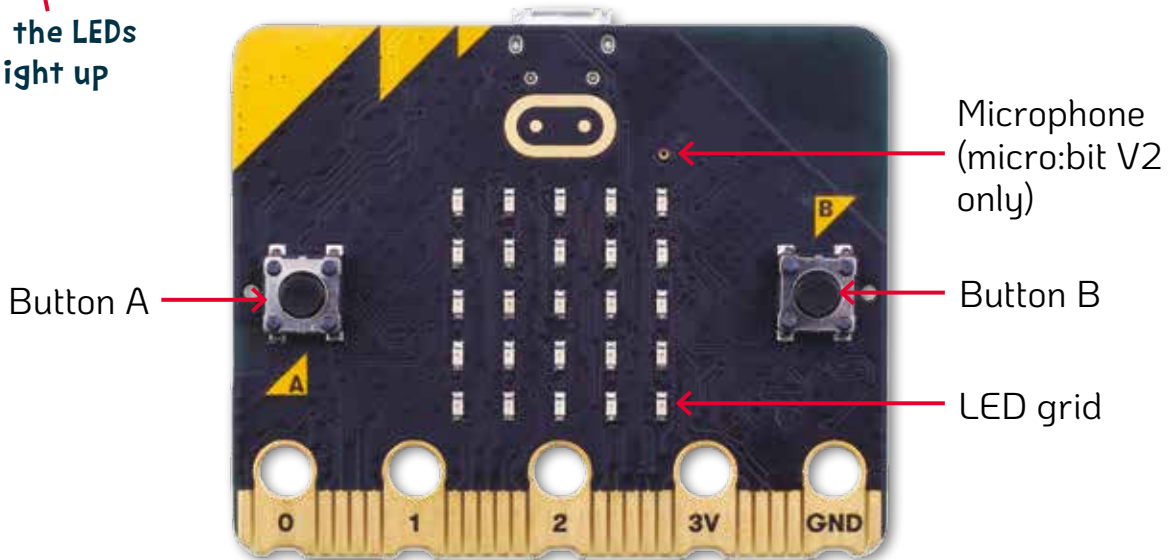
Get to know the micro:bit.



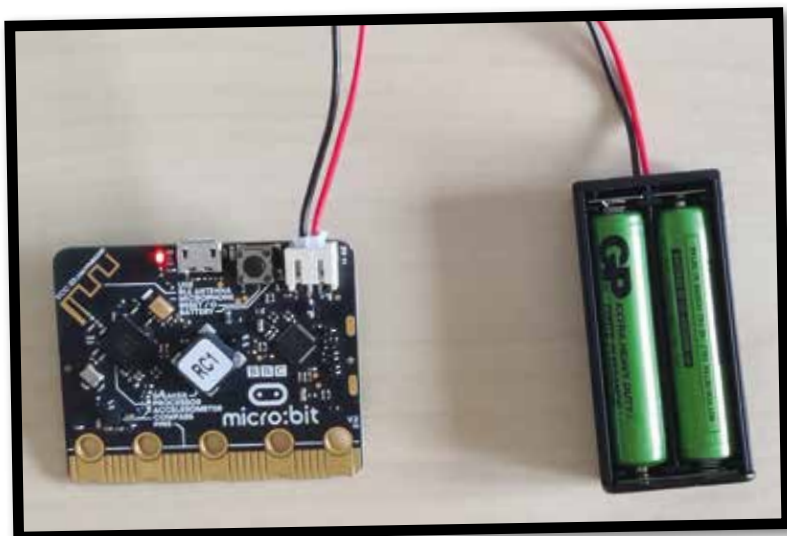
All the LEDs light up

SKILLS YOU'LL LEARN

- Identify micro:bit features
- Locate each component
- Identify what each component does



2 Get connected



Power the micro:bit with the computer or the battery pack.

SKILLS YOU'LL LEARN

- Attach a battery pack
- Attach the microUSB cable



Get
powered
up!

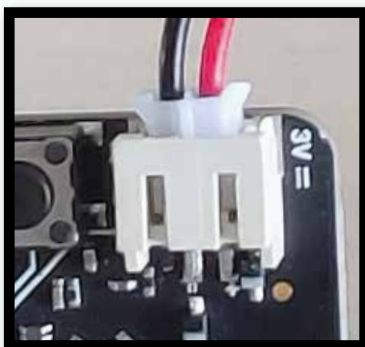
HOW TO DO IT

Connect the microUSB cable and computer



Attach the smaller end of the microUSB cable to the microUSB slot on the micro:bit. Connect the other end of the cable to a computer. You should hear a beep and see the *MICROBIT* folder open on your screen.

Connect the battery pack



Some projects require the micro:bit to be portable so you can use it away from your computer. For these, you'll need to use the battery pack and two AAA batteries.

First, write your program and download it to the micro:bit, then unplug the micro:bit from the computer. Add both batteries to the battery pack and connect it, via the wire connector, to the micro:bit's battery socket. Once attached, the batteries power the micro:bit. Save battery life by removing the pack when you've finished using the micro:bit. Hold the white socket and gently move it from side to side to release it. Do not pull the wire, it might break. **You cannot code the micro:bit while it's running on battery power; it must be connected to a computer.**

3 Access the coding interface

Visit the micro:bit website to get started with coding.

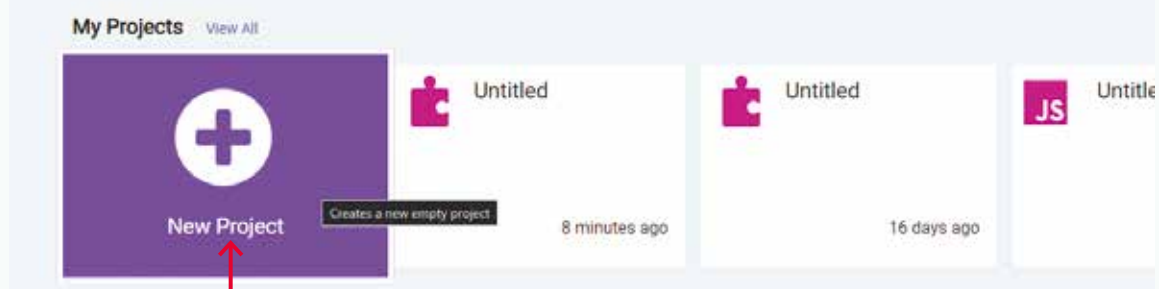
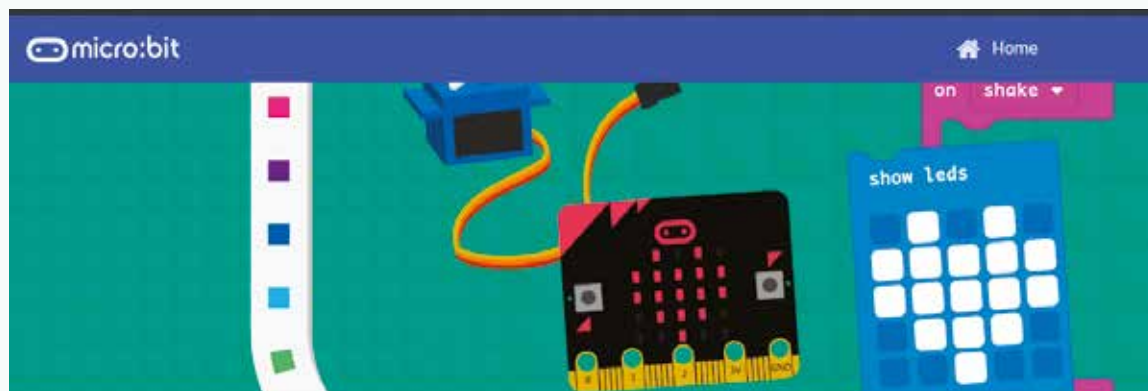
SKILLS YOU'LL LEARN

- Access the micro:bit MakeCode editor

Connect the micro:bit to your computer. Go to <https://microbit.org/>

HOW TO DO IT

Click **Let's code** at the top of the page, then click **Go to MakeCode editor**.



Click **New Project**, name your project, and click **Create** to begin coding.

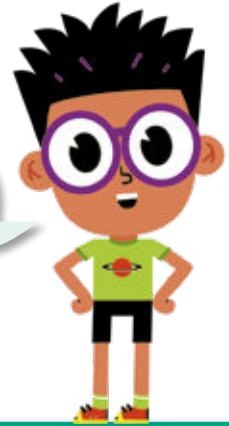
4 Explore the MakeCode editor

Become familiar with the MakeCode editor coding interface.

SKILLS YOU'LL LEARN

- Navigate the coding interface

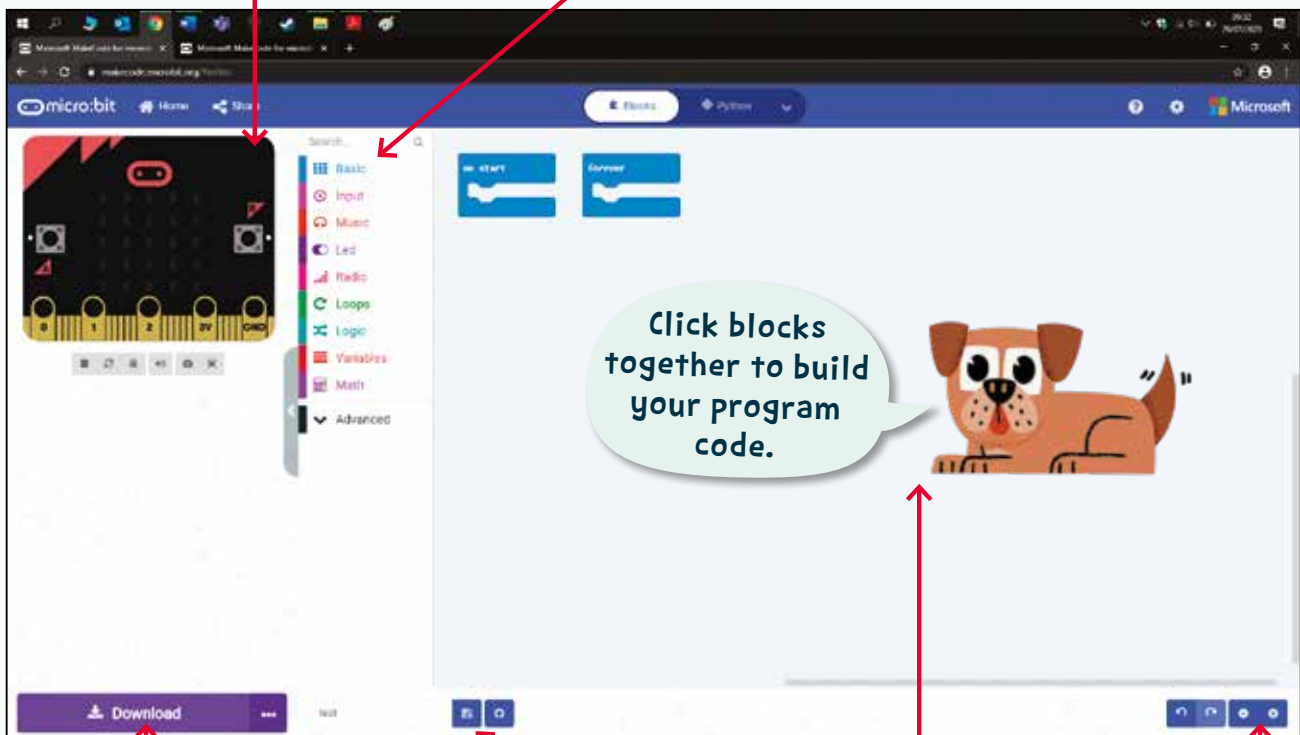
The MakeCode editor is clear and easy to use. Explore its features below.



HOW TO DO IT

Use the simulator to test code

Select from code block groups



Click blocks together to build your program code.

Click to download your code

Click to save your programs

Add blocks to the coding area

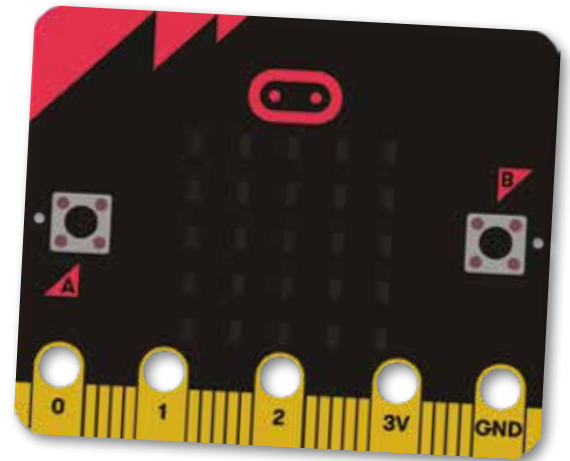
Zoom in or out

5 Start a program

Begin writing your first program for the micro:bit.

SKILLS YOU'LL LEARN

- Write a micro:bit program



GET YOUR BLOCKS

on start

Blocks snap together to help you create code.



PUT IT TOGETHER

The **on start** block appears in the coding area by default, but you can also select it from the **Basic** group.

THEORY

The **on start** block tells the micro:bit to perform the action in the code when the program starts.

6 Show a string

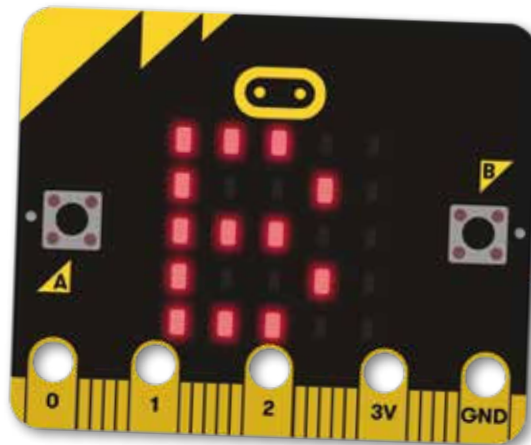
Complete the micro:bit program from the previous step to scroll a custom message across the LED grid.

SKILLS YOU'LL LEARN

- Create a string

WHAT YOU'LL NEED

- Battery pack



GET YOUR BLOCKS

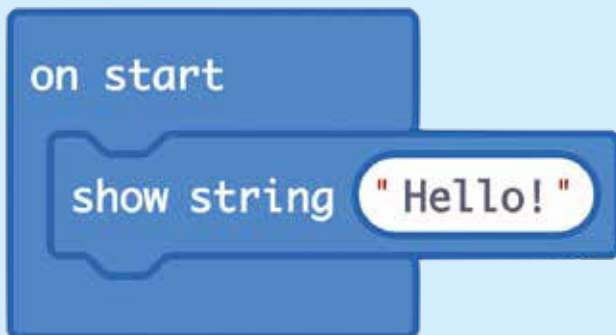
show string "Hello!"

Each letter of the string is scrolled across the display.



PUT IT TOGETHER

In the **Basic** group, click and drag the **show string** block. Attach it to the **on start** block.



Change the string from **Hello!** to your own message.

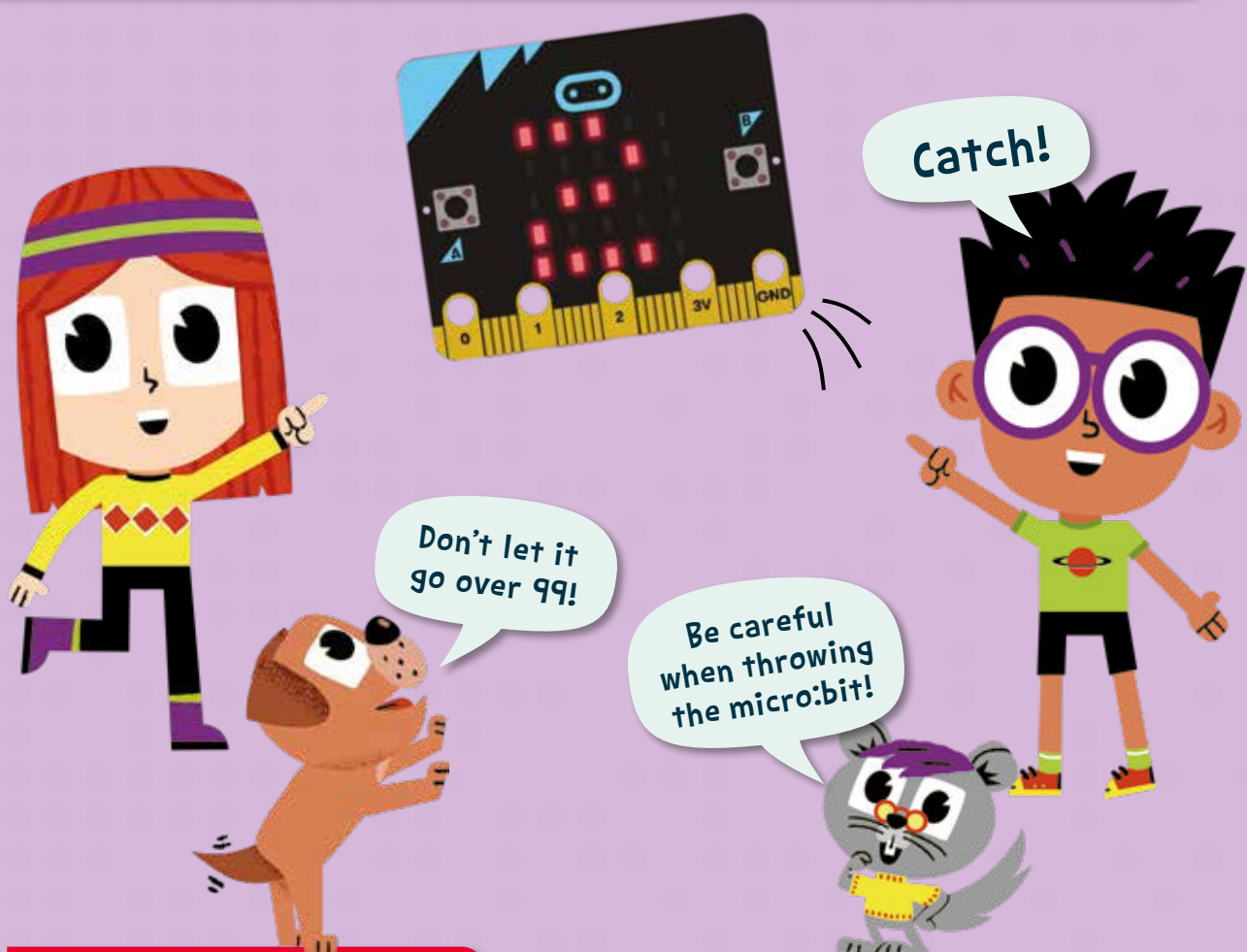
THEORY

The **show string** block scrolls a *string* – a series of text, digits, and symbols – of your choosing across the LED grid.

Project 1 >

The 99 game

In this game of chance, players stand in a circle and toss the micro:bit to each other. Each time someone throws the micro:bit it displays a random number. The player adds this number to a running total, starting from 0. When the total goes over 99, the player holding the micro:bit is eliminated, and the count resets to 0. The game continues until only one player remains.



SKILLS YOU'LL LEARN

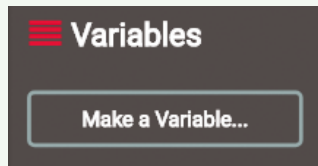
- Create variables
- Select a random number
- Use the accelerometer

WHAT YOU'LL NEED

- Battery pack
- Two AAA batteries
- An elastic or rubber band

1 Create a variable

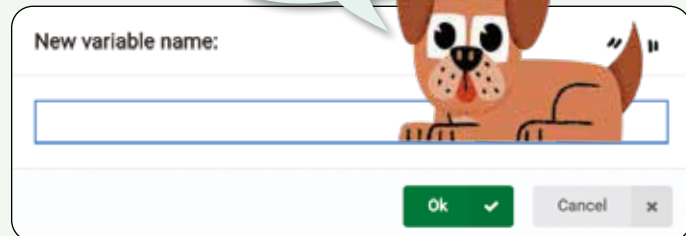
Get your blocks



Type the word 'number' here.

Put it together

In the **Variables** menu, click **Make a Variable**. Name the new variable **number** and click **Ok**.



Theory

When you throw the micro:bit to another player, the program generates a random number and stores it in the **number** variable.

2 Select a random number

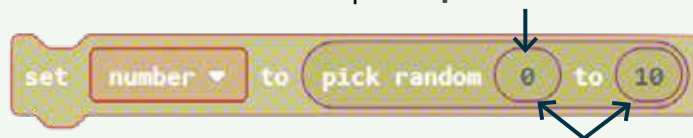
Get your blocks



Put it together

Drag the **set number to** block from the **Variables** group and the **pick random** block from the **Math** group into the coding area.

Drop the **pick random** block here.



Leave 0 and 10 as your range of numbers.

Theory

The program selects a random number between 0 and 10 and stores it in the **number** variable.

Setting a lower range, such as 0 to 5, would make the game last longer.



3 Show the number

Get your blocks



Put it together

Drag a **show string** block from the **Basic** group, and a **number** block from **Variables** into the coding area.

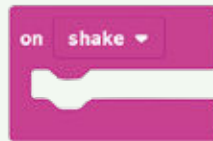


Theory

The program displays the count stored in the **number** variable on the LED grid so each player can see it before it's added to the total.

4 Sense movement

Get your blocks

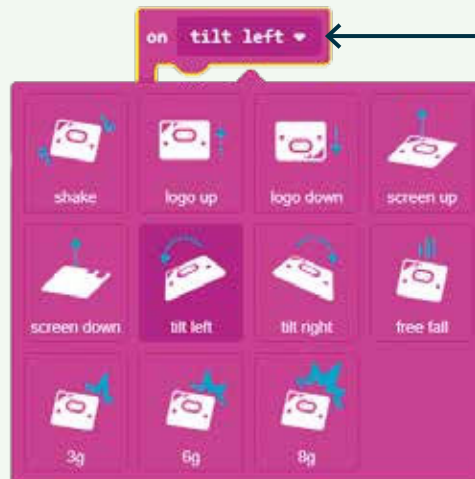


Shake it like a Polaroid picture!



Put it together

Drag the **on shake** block from the **Input** group into the coding area.



Use the drop-down arrow to choose **tilt left**.

Theory

The **Input** block executes the program whenever the micro:bit accelerometer senses the movement you choose. We use **tilt left** so the micro:bit senses when it has been thrown and caught. You can change the movement to suit your gameplay style – for a gentle pass or a harder throw, for example.

5 Finish the code

Put it together

Add the **set number to** block here.

```

on tilt left
  set number to pick random 0 to 10
  show string number
  
```

Add the **show string** block here.

6 Run the program

Download the code onto your micro:bit and add the battery pack. Use a small elastic band to secure the battery pack to the micro:bit.

Start the game!

To clear the display and start a new game, press the **Reset** button.



7 Try this

Program your micro:bit so that it does one of the following:

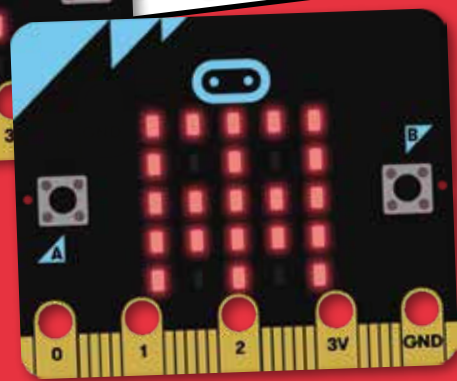
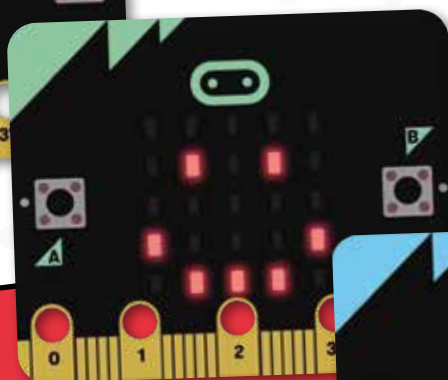
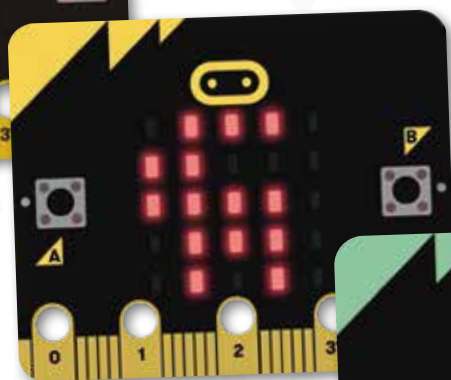
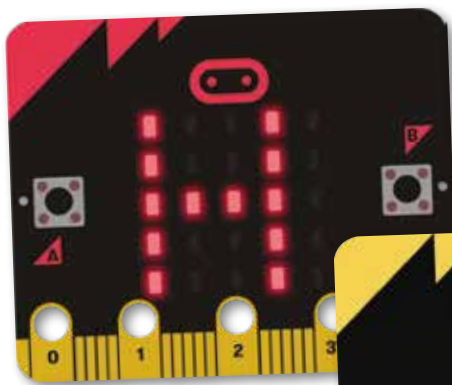
- It starts at 99 and counts down, subtracting a random number each time.
- If you throw a 5, the micro:bit displays an X and the player misses a turn.

The QuestKids® series is designed to make learning fun! Let the two best friends along with their pets inspire you to learn new skills.

Using the proven approach of learning **In Easy Steps**, this guide is:

- Easy to follow
- Fully illustrated
- Packed with tips

The BBC micro:bit is a small but powerful computer that you can learn to code in minutes. This book explains everything you need to know in clear, easy steps. As your skill set grows, you'll create your own cool projects. So whether you're a coding newbie, or you think you know it all, grab your micro:bit and let's get coding!



An engaging and easy introduction to the micro:bit.



You'll learn how to:

- Control the micro:bit with code to access all its hardware
- Code the micro:bit to play music, send messages, detect light, and much, much more
- Create great interactive coding projects using the micro:bit features
- Compete against friends and family with some clever coding games

Supported by a designated website www.thequestkids.com
Learning couldn't be more fun!

£14.99 UK

ISBN 978-1-78791-000-3



CATEGORIES

Children's/Hobbies/Computers
Children's/Education/Technology/IT

www.thequestkids.com | www.ineasysteps.com