

Code Rock Paper Scissors on a micro:bit with Python

Microbit

Beginner

10 Mins

1 Overview

Can you beat a micro:bit at rock, paper, scissors? In this project you're going to find out, by coding your very own game on a micro:bit!

Give your micro:bit a shake and the accelerometer inside it feels the movement. Your code then picks rock, paper or scissors at random and shows it as a little picture on the LED matrix (the grid of lights on the front), so you can see straight away whether you've won.

The best bit? When you're done you'll have a real game you can play against your micro:bit, or challenge a friend. Don't worry if something doesn't work first time, that's all part of coding. Let's give it a go!



What you'll learn

- Bring in a ready-made library with importing libraries
- Use a while loop to keep your game running
- Make and use your own variable
- Use the micro:bit's shake sensor to spot movement
- Draw your own pictures on the grid of lights
- Make decisions with if statements, plus elif and else
- Compare two things with a comparison operator
- Let the micro:bit choose at random

2 What you'll need



What you'll need

- The micro:bit Python editor
- micro:bit
- micro USB cable
- battery pack for the micro:bit (optional)

3 Key words



Key words

Importing Libraries

A library is a ready-made bundle of code that someone else has written for you. Importing a library means bringing it into your project so you can use all its ready-made commands and features, without building them from scratch.

It's like borrowing a recipe book from the library. Instead of working out every recipe yourself, you bring the book in and use the recipes whenever you need them.

Variable

A variable is something that can change. Think of it like a labelled box: you pop something inside, and the label tells you what it is. You can swap what's inside whenever you like.

A variable called score might start at 0, then change to 1, 2 and 3 as you collect points in a game.

While loop

A while loop keeps repeating its instructions for as long as something stays true. The moment that thing stops being true, the loop stops.

"While it's still raining, keep the umbrella up." As soon as the rain stops, you put the umbrella down, and the loop ends.

Accelerometer

An accelerometer is a tiny part inside the micro:bit that senses movement, like tilting, shaking or being turned over. It turns that movement into numbers your code can react to.

A bit like the sensor in a games controller that knows when you tilt it to steer, the accelerometer notices when the micro:bit moves.

Random

Random means you can't guess what's coming next, a bit like rolling a dice. The computer picks for you, so you get a different surprise each time.

Asking a computer for a random number between 1 and 6 is just like rolling a dice, you never know if you'll get a 1 or a 6.

If statement

An if statement checks whether something is true, and only runs its instructions when it is. It's how code makes decisions.

"If it's cold, put on a coat." If it's warm, you skip the coat, the action only happens when the check is true.

Comparison Operators

A comparison operator checks how two things measure up, for example whether they are equal or one is bigger than the other. It gives back a simple true or false answer.

Asking "is 6 == 6?" gives back true, but "is 6 == 2?" gives back false, so your code knows whether the two numbers match.

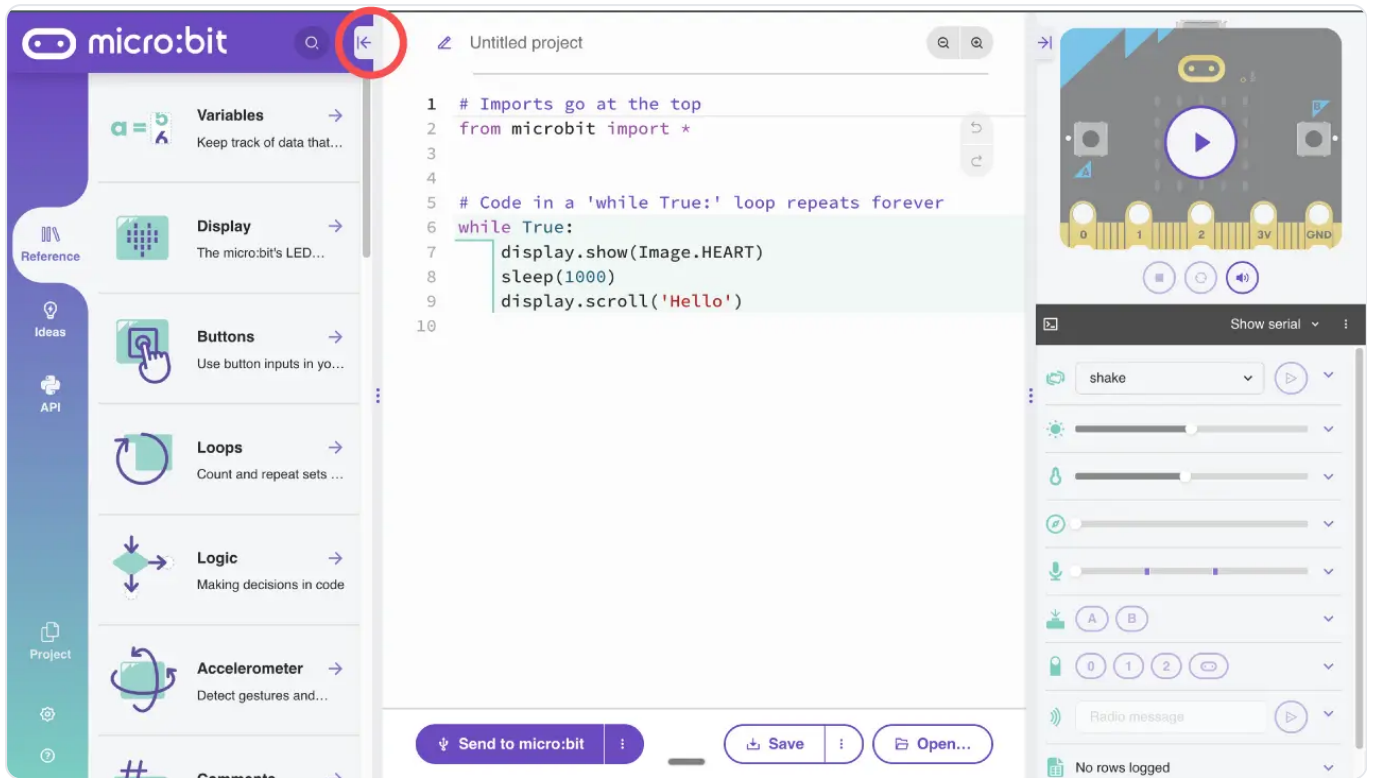
LED Matrix

The LED matrix is the grid of little red lights on the front of the micro:bit. By switching them on and off in patterns, your code can show numbers, letters and tiny pictures.

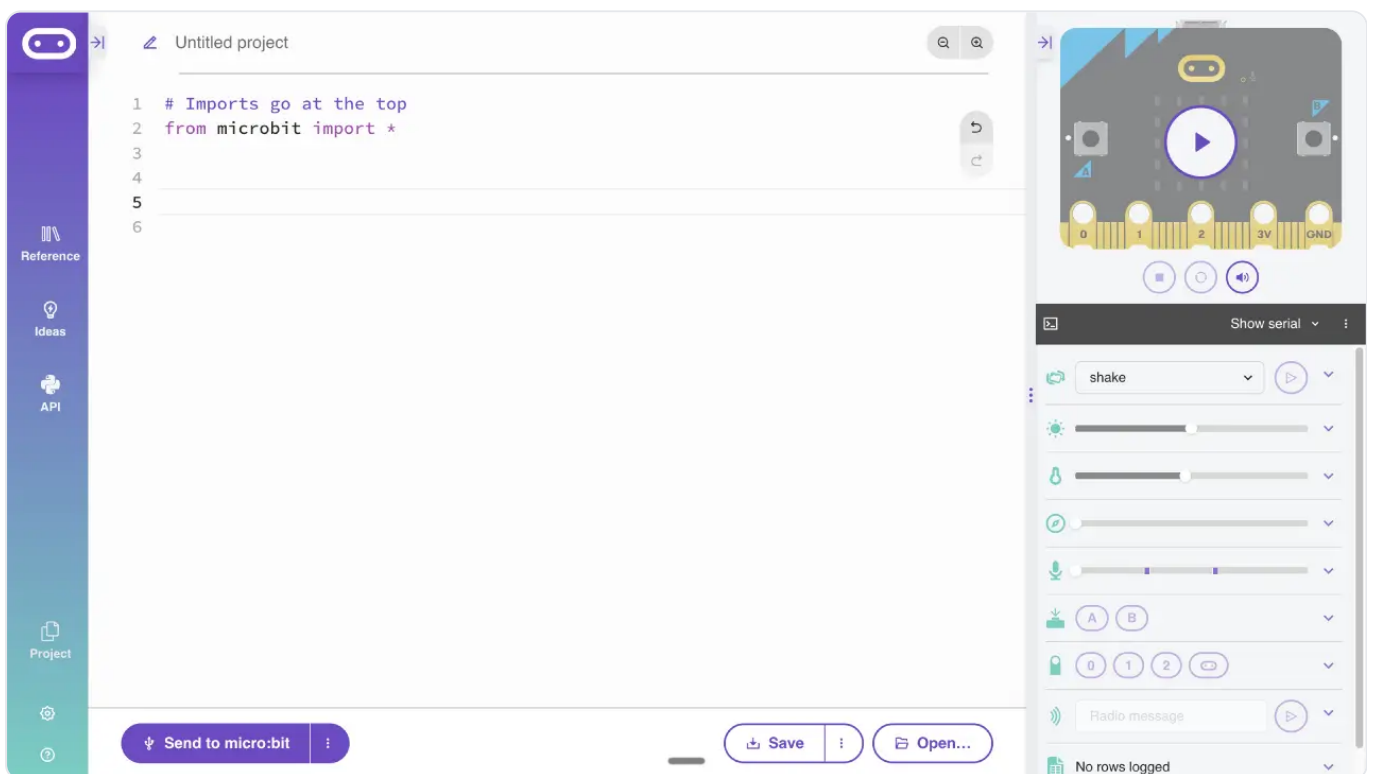
Like a mini scoreboard made of dots, the LED matrix lights up to spell out your step count.

4 Set up your micro:bit Python Editor

1. Open your web browser. We recommend Google Chrome or Microsoft Edge.
2. In the address bar type python.microbit.org.
3. Give your project a name: select the project name box at the top of the editor and type **Rock Paper Scissors**.
4. Close the left-hand panel by selecting the **arrow** pointing to the left.



5. Delete the code on lines **5 to 9** in the main code area.



5 Import the random library

On line 3, type `import random`. This is importing libraries, and it brings in the random library so the micro:bit can pick for itself later on.

```

1 # Imports go at the top
2 from microbit import *
3 import random
4

```

6 Create a while True loop

Type `while True:` and press **Enter**. This creates a while loop, so anything inside it runs while the condition stays true. Python needs the capital **T** in `True` and the colon at the end, or the code won't run. Notice your cursor automatically indents, that's how Python shows what's inside the loop.

```

1 # Imports go at the top
2 from microbit import *
3 import random
4 while True:
5

```

7 Create the picture variables

Each of these makes a variable holding one of your pictures.

1. Type `paper = Image("99999:90009:90009:90009:99999")` to create the **paper** icon.
2. Type `rock = Image("00000:09990:09990:09990:00000")` to create the **rock** icon.
3. Type `scissors = Image("90099:09099:00900:09099:90099")` to create the **scissors** icon.

The **9**s are the lights that switch on and the **0**s are the lights that stay off, so each row of numbers draws one row of the grid.

```

1 # Imports go at the top
2 from microbit import *
3 import random
4 while True:
5     paper = Image("99999:90009:90009:90009:99999")
6     rock = Image("00000:09990:09990:09990:00000")
7     scissors = Image("90099:09099:00900:09099:90099")
8

```

8 Detect a shake

1. Type `if accelerometer.was_gesture("shake"):` and press **Enter**. Keep it lined up with your picture variables. This `if` statement checks whether the micro:bit has been shaken.
2. Type `choice = random.randint(0, 2)` and press **Enter**. This creates a **choice** variable and sets it to a random whole number between **0** and **2**.

```
1  # Imports go at the top
2  from microbit import *
3  import random
4  while True:
5      paper = Image("99999:90009:90009:90009:99999")
6      rock = Image ("00000:09990:09990:09990:00000")
7      scissors = Image ("90099:09099:00900:09099:90099")
8      if accelerometer.was_gesture("shake"):
9          choice = random.randint(0,2)
10
```

9 Show paper with an if statement

1. Type `if choice == 0:` and press **Enter**. The `==` is a comparison operator, and it checks whether the two things match.
2. Type `display.show(paper)` and press **Enter**.
3. Delete the indent so your cursor lines back up with the `if choice == 0:` line.

Now, when **choice** is **0**, the **paper** icon appears on the LED matrix.

```
1  # Imports go at the top
2  from microbit import *
3  import random
4  while True:
5      paper = Image("99999:90009:90009:90009:99999")
6      rock = Image ("00000:09990:09990:09990:00000")
7      scissors = Image ("90099:09099:00900:09099:90099")
8      if accelerometer.was_gesture("shake"):
9          choice = random.randint(0,2)
10         if choice == 0:
11             display.show(paper)
12
```

10 Show rock with an elif statement

1. Type `elif choice == 1:` and press **Enter**. `elif` is short for "else if", so this is the next check to try.
2. Type `display.show(rock)` and press **Enter**.
3. Delete the indent so your cursor lines back up with the `elif choice == 1:` line.

Now, when **choice** is **1**, the **rock** icon appears instead.

```
1  # Imports go at the top
2  from microbit import *
3  import random
4  while True:
5      paper = Image("99999:90009:90009:90009:99999")
6      rock = Image ("00000:09990:09990:09990:00000")
7      scissors = Image ("90099:09099:00900:09099:90099")
8      if accelerometer.was_gesture("shake"):
9          choice = random.randint(0,2)
10         if choice == 0:
11             display.show(paper)
12         elif choice == 1:
13             display.show(rock)
14
```

11 Show scissors with an else statement

1. Type `else:` and press **Enter**.
2. Type `display.show(scissors)` .

`else` is the catch-all, so if **choice** isn't **0** or **1** it must be **2**, and the **scissors** icon appears.

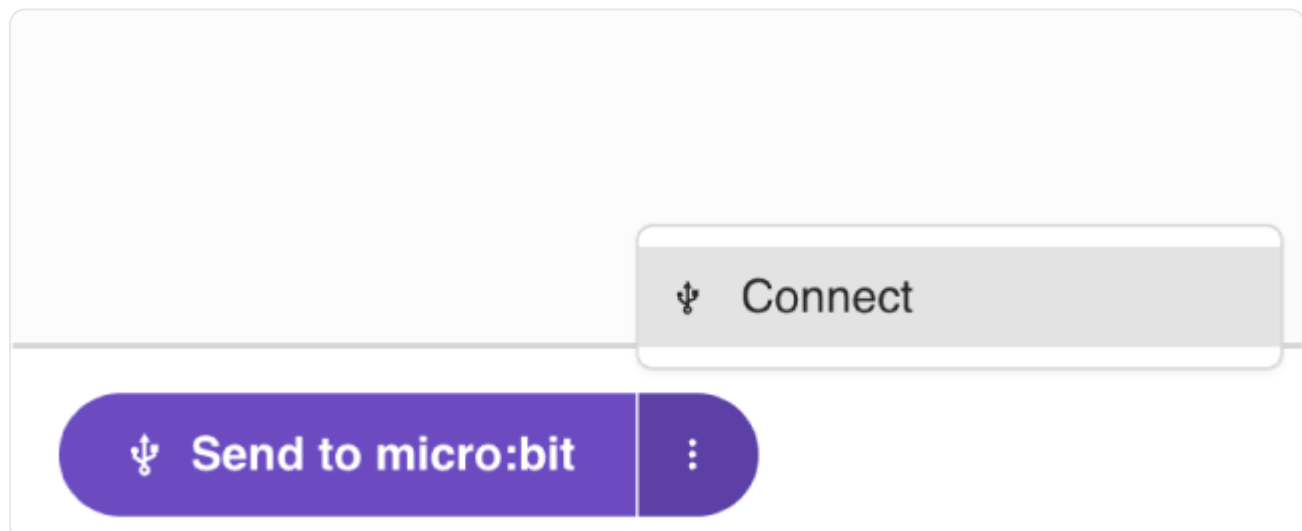
```

1  # Imports go at the top
2  from microbit import *
3  import random
4  while True:
5      paper = Image("99999:90009:90009:90009:99999")
6      rock = Image ("00000:09990:09990:09990:00000")
7      scissors = Image ("90099:09099:00900:09099:90099")
8      if accelerometer.was_gesture("shake"):
9          choice = random.randint(0,2)
10         if choice == 0:
11             display.show(paper)
12         elif choice == 1:
13             display.show(rock)
14         else:
15             display.show(scissors)

```

12 Downloading Your Code

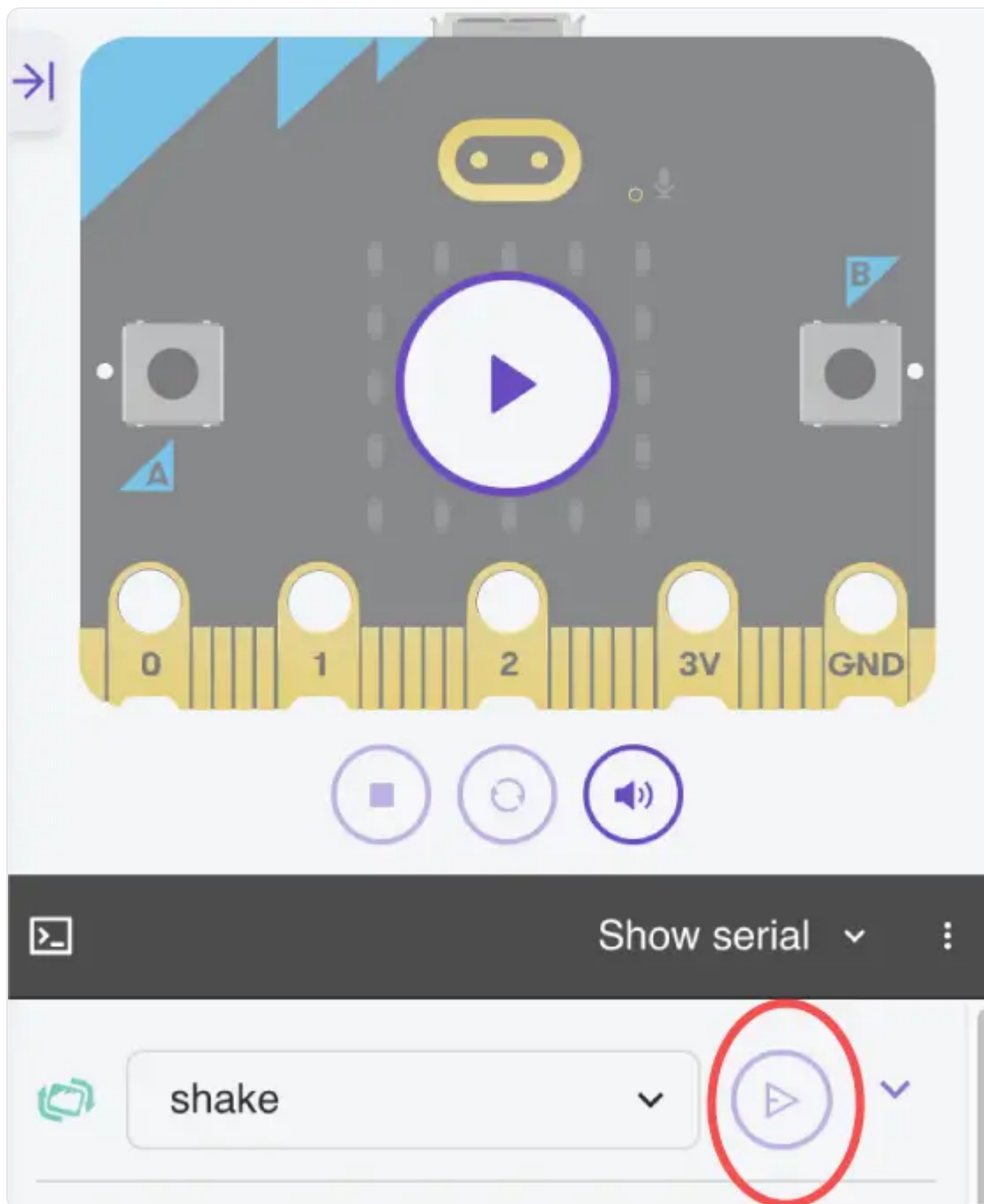
1. Connect the micro:bit to the computer with the micro USB cable.
2. Select the **three dots** next to **Send to micro:bit**.
3. Select **Connect** and follow the on-screen prompts.
4. Select **Send to micro:bit** to download the code.



13 Run it and Play

In the web browser

The Python editor has a built-in micro:bit simulator, handy if you haven't got a micro:bit to hand. Below the simulator you'll see the **shake** function is already selected. Select the **play** button next to it to simulate a shake and watch paper, rock or scissors appear on the LED matrix.



On a real micro:bit

Once your code's downloaded, give the micro:bit a shake and see what it picks. Remember the choice is random, so you might get the same icon twice in a row, that's completely normal. Why not find a friend and see who can beat the micro:bit?

14 Try it yourself

Try it yourself

Challenge: Give your micro:bit a voice! Add an `audio.play(Sound.HAPPY)` line inside your `if accelerometer.was_gesture("shake"):` check, just after you set the choice, so it beeps every time it plays. For an extra challenge, pick a different sound for rock, paper and scissors.

Good to know: the sound plays through the built-in speaker, which only the micro:bit V2 has. If you've got a V1, you can still try this by clipping a buzzer or headphones to the pins. Everything else in this project works happily on both.

15 Stuck? Quick fixes

Problem	Try this
Nothing happens when I shake it	Give the micro:bit one firm, clear shake, not too gentle. Also check your code is indented inside the <code>if accelerometer.was_gesture("shake"):</code> line.
The same icon keeps showing	That's okay, it's working! The choice is random, so shake a few more times and the others will turn up.
My icons look wrong or blank	Check each <code>Image("...")</code> line and make sure the 9s and 0s match the pattern for rock, paper and scissors, with a colon between each row.
It only ever shows two icons	Check your <code>elif</code> is set to <code>choice == 1</code> , and that <code>random.randint</code> goes from 0 to 2 .
I get an error about <code>random</code>	Check <code>import random</code> is on its own line at the top, above your <code>while True:</code> loop, and spelled all in lower case.
My code won't download to the micro:bit	Use a data USB cable (not a charge-only one), make sure the micro:bit is paired, and try a different USB port.