

Introduction to the BBC micro:bit

TechResort Mini Makers Session #1119



What do I need?

- A laptop with internet access
- A BBC micro:bit
- A USB cable to connect the two together
- A mouse (optional)

What is the micro:bit?

This is a tiny programmable computer which children will soon be using at school to introduce them to coding and electronics.



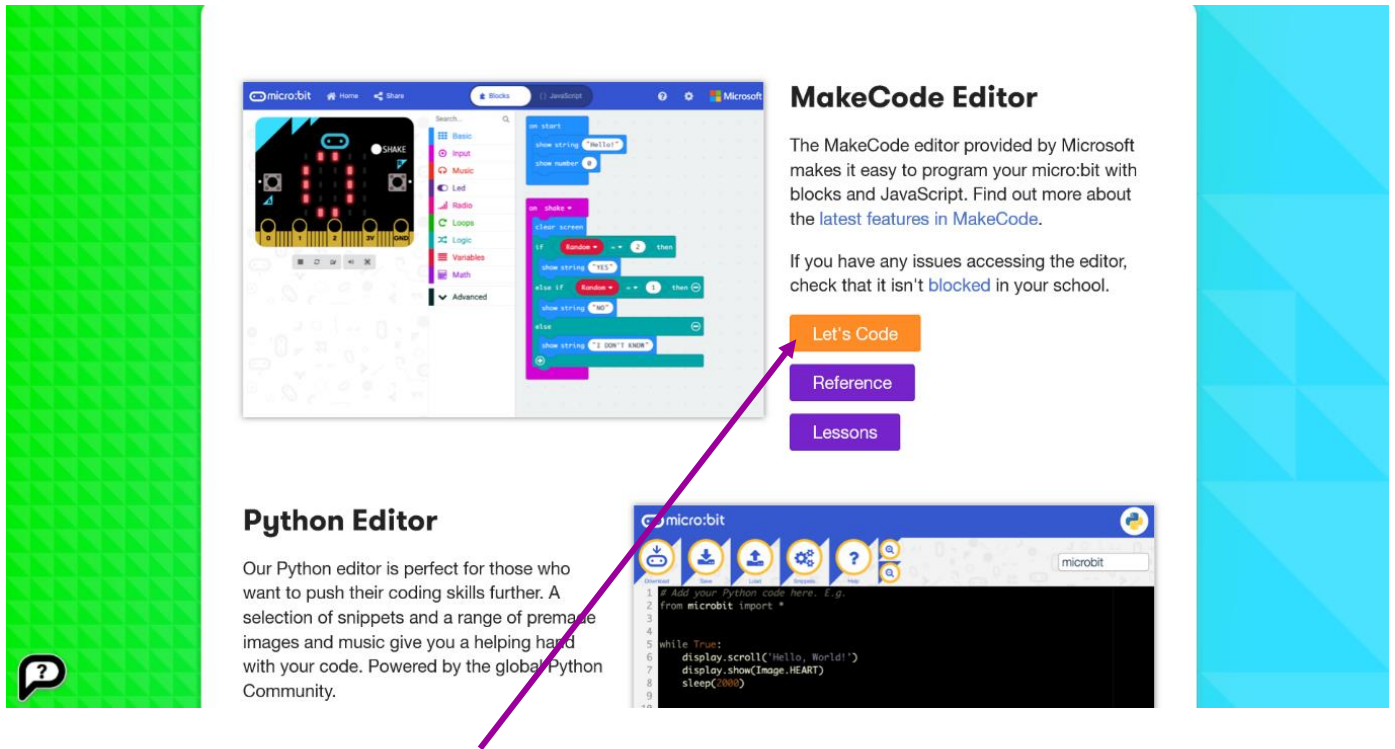
How are we going to be using the micro:bit?

Today you will learn how to write programs for the micro:bit and how to run them on your computer.

Once you have finished reading this page, please turn it over and begin following the script. Please ask an elf if you need help.

Getting Started

- On your computer, go to <https://microbit.org/code/> - you will see that there are a couple of different coding environments.



The image shows a screenshot of the microbit.org website. On the left, there is a green vertical bar with a white question mark icon. The main content area is divided into two sections. The top section is titled "MakeCode Editor" and features a screenshot of the MakeCode editor interface with a block-based code editor. To the right of this screenshot is a text block describing the editor and a "Let's Code" button. Below this are two purple buttons labeled "Reference" and "Lessons". The bottom section is titled "Python Editor" and features a screenshot of the Python editor interface with a text-based code editor. A purple arrow points from the "Let's Code" button to the Python Editor section.

MakeCode Editor

The MakeCode editor provided by Microsoft makes it easy to program your micro:bit with blocks and JavaScript. Find out more about the latest features in [MakeCode](#).

If you have any issues accessing the editor, check that it isn't [blocked](#) in your school.

[Let's Code](#)

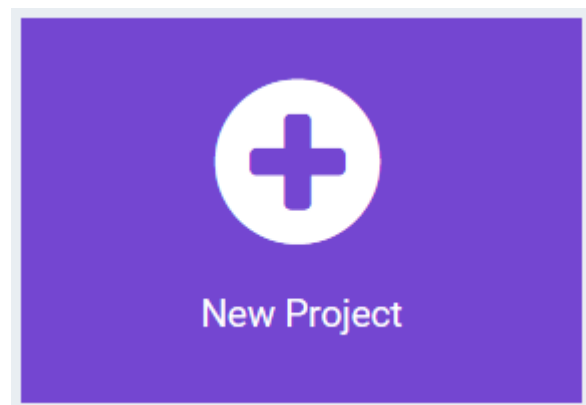
[Reference](#)

[Lessons](#)

Python Editor

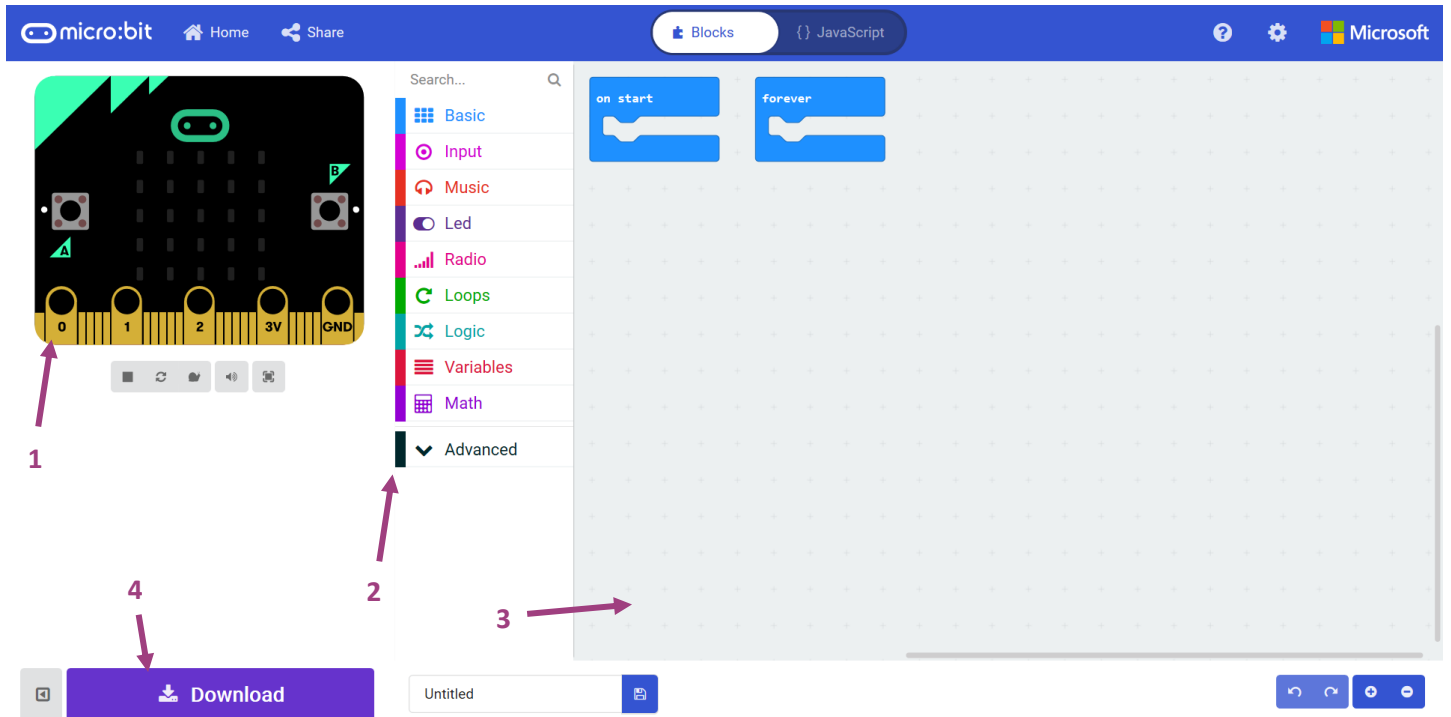
Our Python editor is perfect for those who want to push their coding skills further. A selection of snippets and a range of pre-made images and music give you a helping hand with your code. Powered by the global Python Community.

- Click on **Let's Code** under the 'MakeCode Editor' section.
- Next click on the 'New Project' button:



Introducing the MakeCode Editor

Now that the editor is open, you can see where you will place the code. MakeCode Editor is deliberately set out to look similar to Scratch, so that you are immediately familiar with the interface.



1. When you change your code, the image of the micro:bit on the far left will show you what your code will do.
 2. To the right of that (in the middle) are the block types – but you aren't going to use all of them today!
 3. On the right is a grey area that you can drag, drop, and link pieces of code together in order to create a script.
 4. 'Download' will download your code so you can put it on your actual micro:bit.
- There are some buttons under the micro:bit image that can help you too!



This button will stop your code from running and show you...



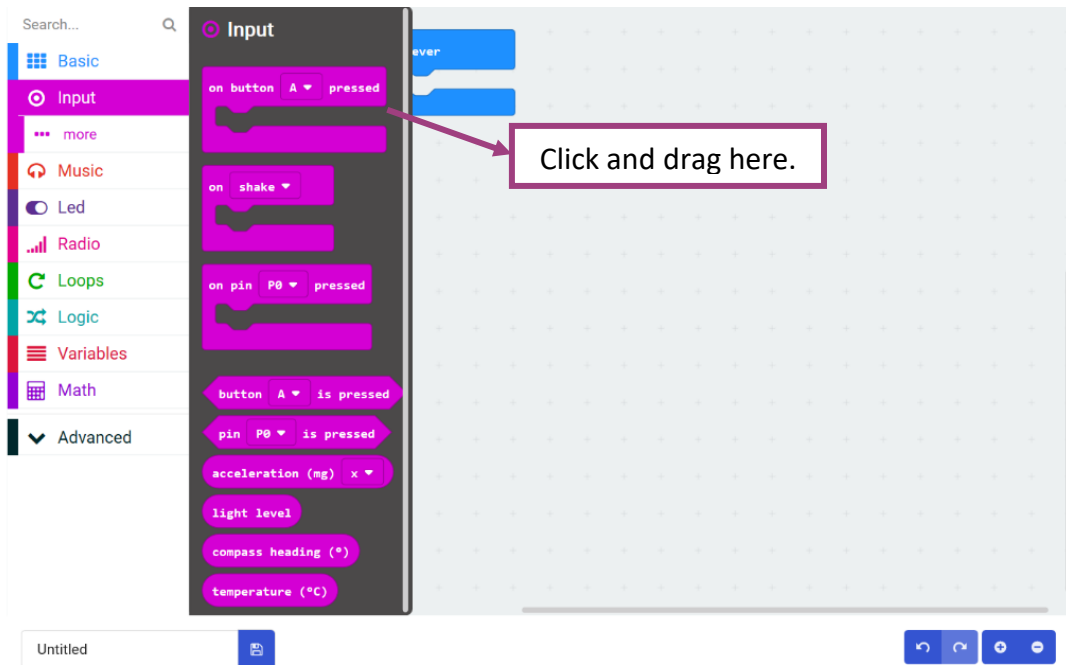
... this button! Which will start your code up again.



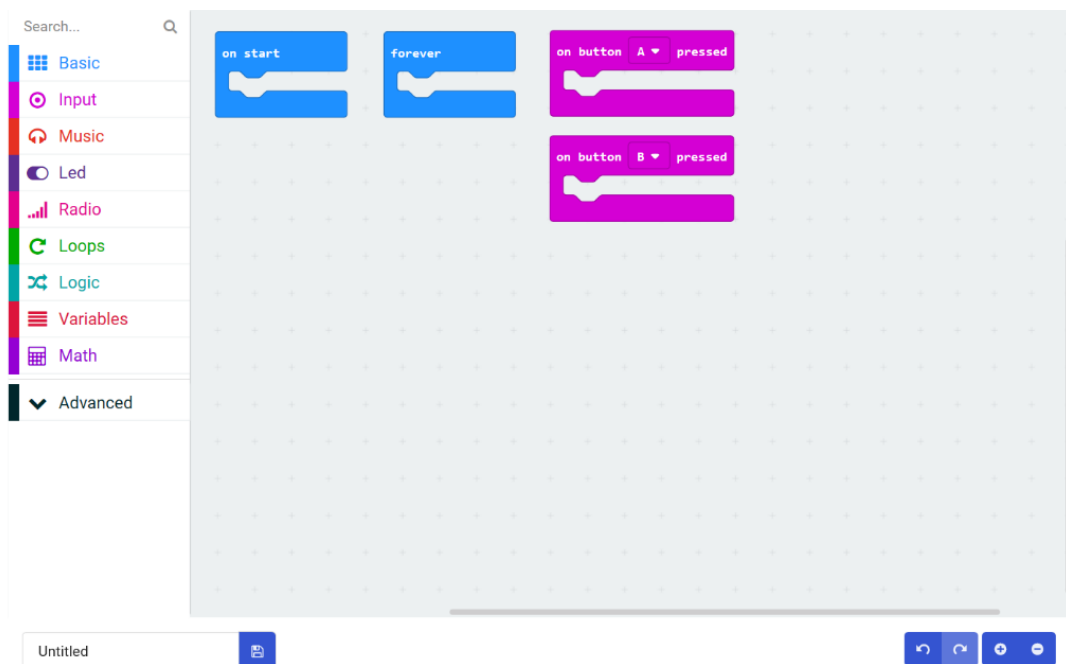
This button will refresh your code.

Your First micro:bit Code

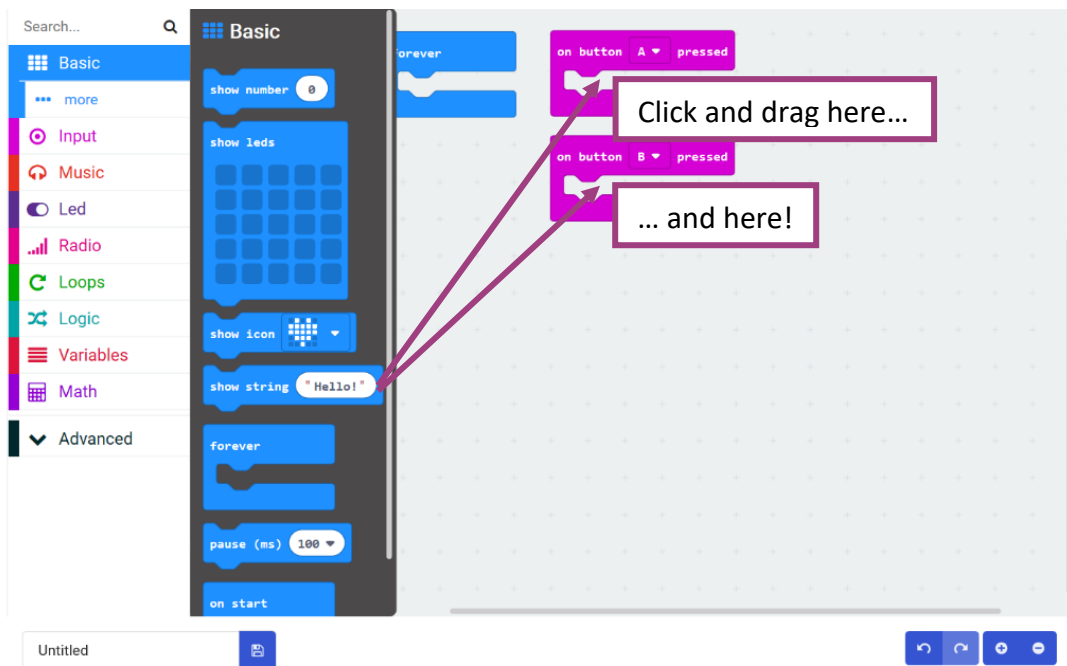
- Click on 'Input' in the middle of the screen, then click and drag the 'on button A pressed' icon into the grey area.



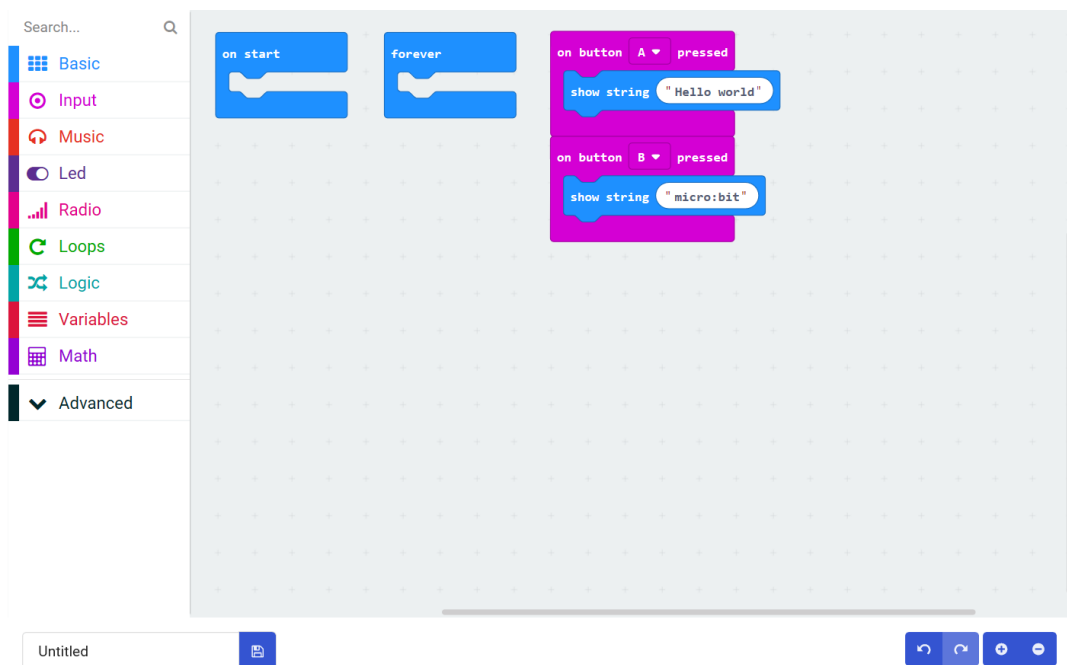
- Repeat this, but this time change the 'A' to a 'B' by clicking on the little white down arrow next to it.
- Your code should now look like this:



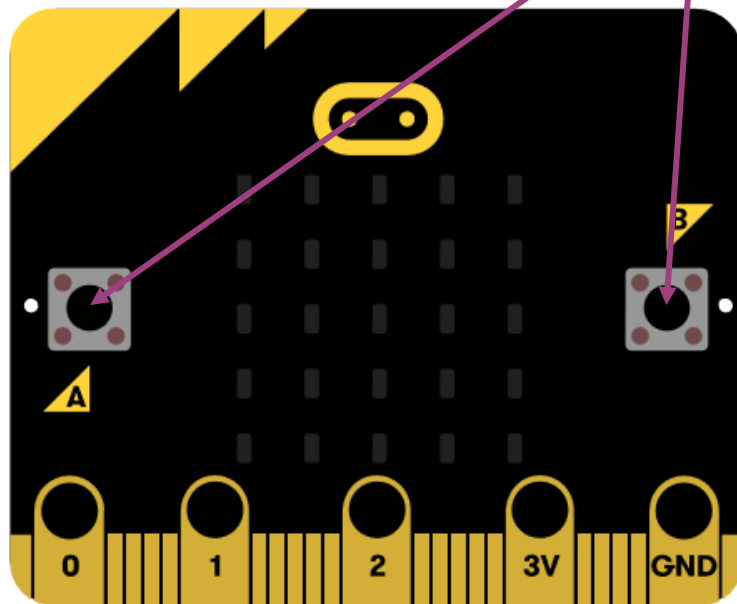
- Now click on 'Basic' in the middle of the screen, then click and drag the 'show string "Hello!"' block into your two pieces of code.



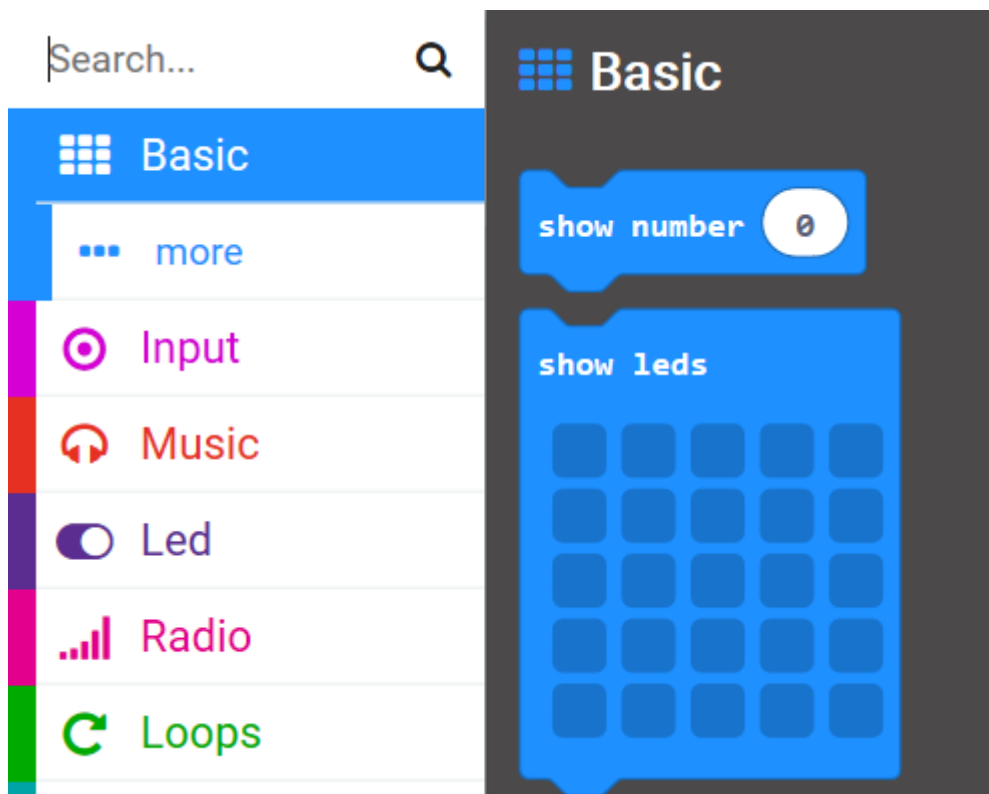
- Now change Button A's string from "Hello!" to say "Hello world", and Button B to say "micro:bit". Your code should now look like this:



- Now look at the image of the micro:bit to see what your code is doing. Nothing? Try clicking on the buttons labelled 'A' and 'B' in the image:




- Do you get the right messages when you click on each button? Congratulations! It looks like your first program is a success!
- Try editing your code to show something different when you press a button. The 'show number' or 'show leds' blocks in the 'Basic' group might be useful here:



How to Get Code Onto Your micro:bit

Getting your code to run on your real micro:bit is much more fun than just running it on a web page!

- First, you might want to give it a name using the text box on the bottom of the screen:

- Next we need to download the code, just click on the 'Download' button at the bottom of the page:



- Very shortly, the image below should appear telling you how to download your code onto your micro:bit.

Don't worry – we will explain how to do this in more detail on the following pages.

Download to your micro:bit



- 1 Connect the micro:bit to your computer with a USB cable**
Use the microUSB port on the top of the micro:bit

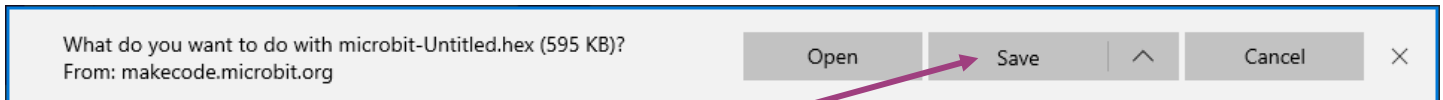


- 2 Move the .hex file to the micro:bit**
Locate the downloaded .hex file and drag it to the MICROBIT drive

Help



- Depending on which browser you are using you may see this message underneath:

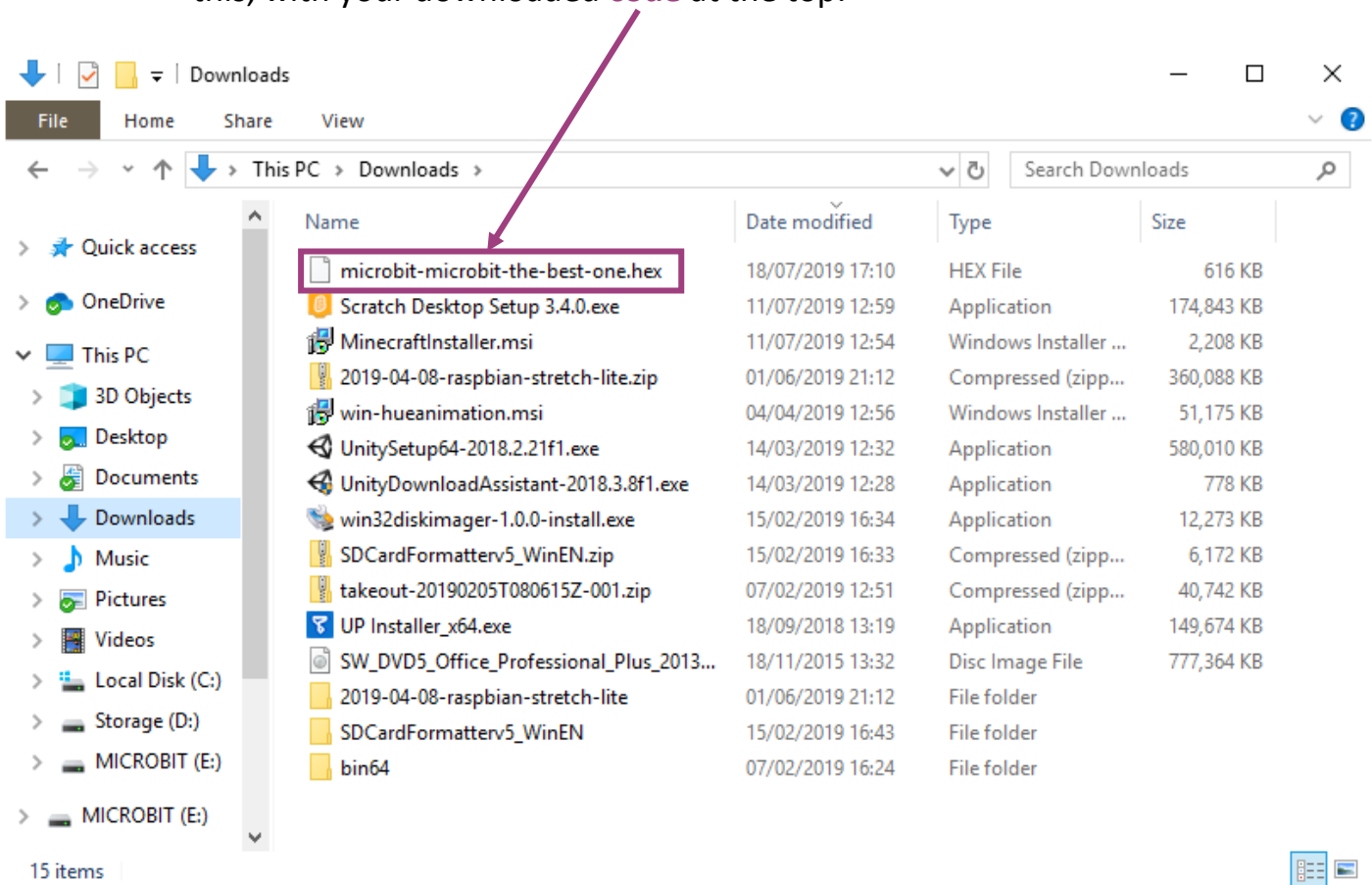


- In which case press **'Save'**.
- If you are using Chrome, it should download automatically.

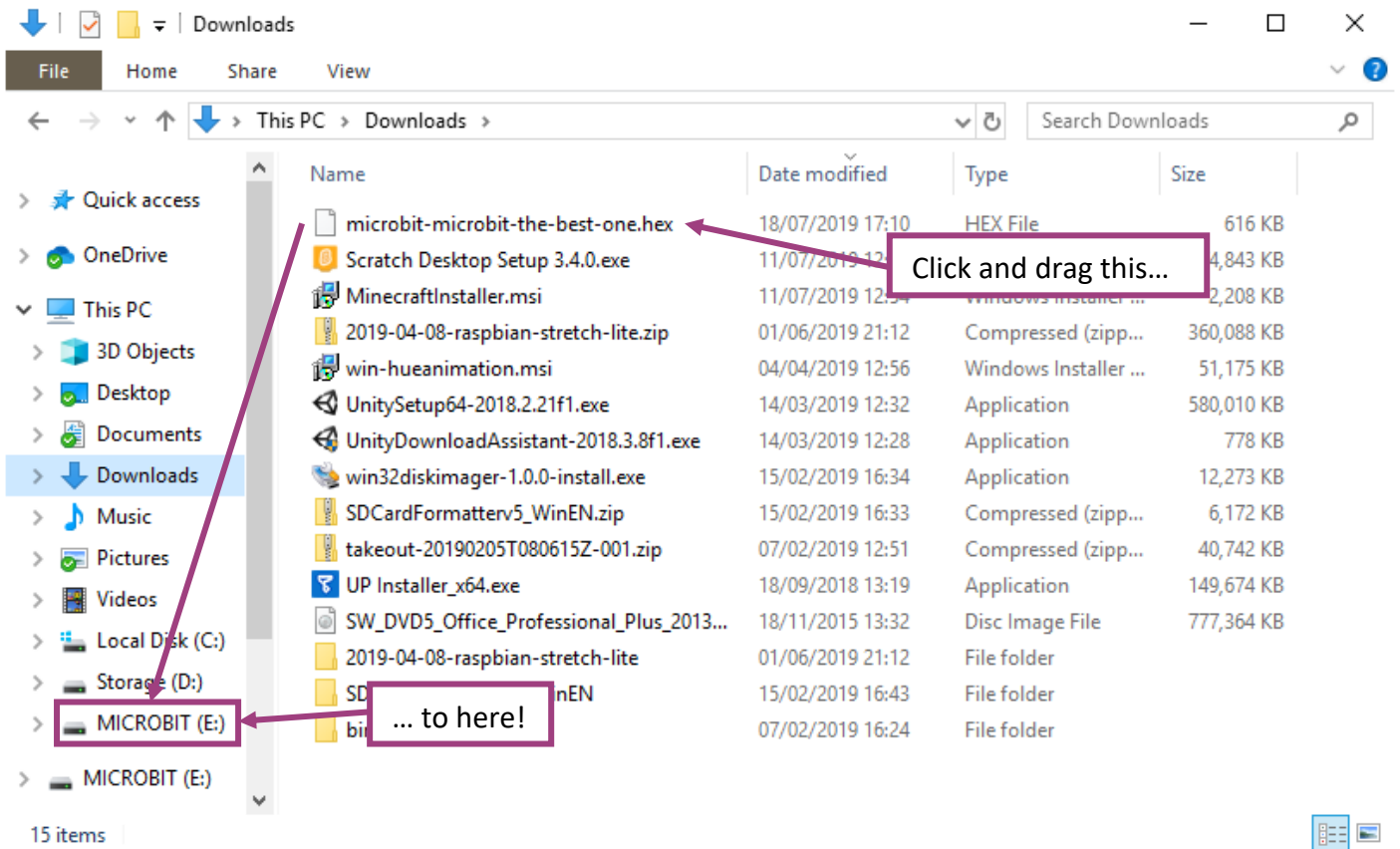
- Now open Windows Explorer by clicking on the folder icon on the Taskbar at the bottom of your screen:



- Find the 'Downloads' folder on the left and click on it to see a list of files. It may be under 'Quick Access' or 'This PC'. It should look something like this, with your downloaded **code** at the top:



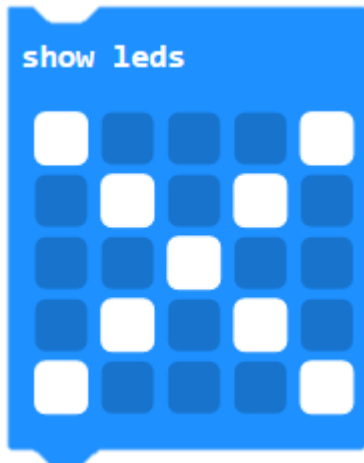
- Now plug your micro:bit into your PC using a USB cable. Somewhere below the Downloads folder (you may need to scroll down) you should see something like MICROBIT (D:) or MICROBIT (E:) listed.
- To program your micro:bit, simply drag your downloaded file onto it like this:



- Copying will take a few seconds, but when it's done try pressing buttons 'A' and 'B' on your real micro:bit to see what happens!
- Good, eh?

Making More Programs on the micro:bit

Using the 5x5 LED co-ordinate system, a pattern can be quickly created. In the basics section, there is a block of code called 'show leds' which you may have used earlier. Using this, you can simply tick the boxes to create a pattern:



- Tick and cross: Make some code so that when the A button is pressed, the LEDs will display a tick. When the B button is pressed, the LEDs will display a cross. Compile the code and copy it over to your BBC micro:bit.
- If you're enjoying this, try some more programs to show other things, for example your initials or happy and sad faces!

Rock-Paper-Scissors Game

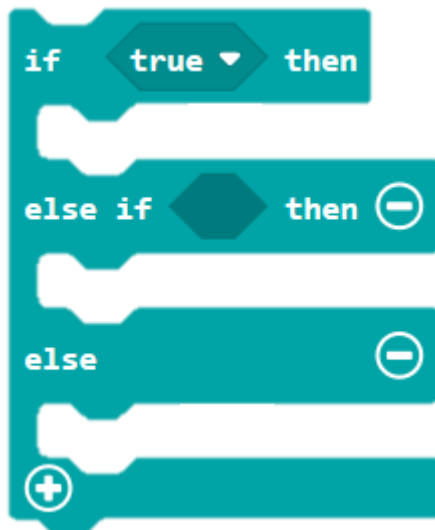
We are now going to create an app that will let us play a game of Rock-Paper-Scissors. You can then play this with friends who also have micro:bits and keep score to see who wins the most games!

- We want the micro:bit to choose rock, paper, or scissors when it is shaken. Let's begin by creating an 'on shake condition' Input so the micro:bit will run code when it is shaken.
- Next, we **make a variable** and store a random number from 0 to 2. When the micro:bit is shaken, the number 0, 1 or 2 will be picked at random. Create the code that looks like this:



- Hint: the colour of the block will show you where to find it on the centre menu. You may also need to **make a variable** first!

- Now we want to tell the micro:bit what to display depending on the value of the random number. To do this we use an 'if; else-if; else' block. Find the 'if <true> then; else' block (hint: it's in the 'Logic' section):
- Now click on the '+' button on the bottom left of the block to increase the number of statements. You should have a block that looks like this now:



- If you end up with too many 'else if' statements, you can press the '-' button to remove some.

- Next, drag your 'if; else-if; else' statement onto your 'on shake' loop, like this:

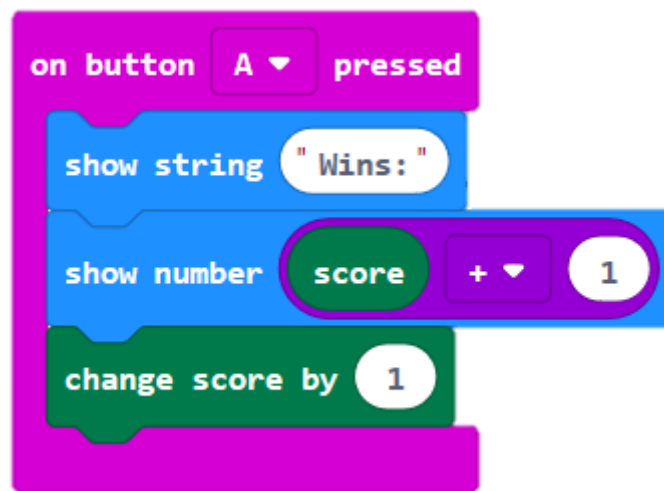


- Now edit your code to look like this. It tells the micro:bit to display a rock if the number is 0, some paper if the number is 1, or a pair of scissors if the number is 2.

```
on shake
  set random_shake to pick random 0 to 2
  if random_shake = 0 then
    show leds
    [Rock pattern]
  else if random_shake = 1 then
    show leds
    [Paper pattern]
  else
    show leds
    [Scissors pattern]
```

The image shows a Scratch code block for a micro:bit. It starts with an 'on shake' event. A 'set random_shake to pick random 0 to 2' block is added. Below this is an 'if random_shake = 0 then' block containing a 'show leds' block with a rock pattern (a 5x5 grid of LEDs with the top row and corners lit). This is followed by an 'else if random_shake = 1 then' block with a 'show leds' block and a paper pattern (a 5x5 grid with the top row and the two middle columns lit). Finally, an 'else' block contains a 'show leds' block with a scissors pattern (a 5x5 grid with the top row, the two middle columns, and the bottom row lit). A plus sign is visible at the bottom of the code block, indicating it can be expanded.

- Now download your code to your micro:bit. Shake it several times. Do you get each of the three images at random?
- Now you can play Rock/Paper/Scissors with anyone else that's programmed their micro:bit the same. Remember: Paper beats Rock, Scissors beats Paper, and Rock beats Scissors!
- How about creating an additional script to keep score? If you win, pressing button 'A' will increase your score by 1. In the 'Game' set of blocks you'll find a variable called 'score'. Add the following to your script:



- Now download your program again. Press button 'A' every time you win. Does it work?
- Can you add a counter to display how many times you have lost when you press button B?

More Fun and Games

- The 'Let's Code' page on the micro:bit website contains lots of tutorials for you to create more games and tools.
- You can also find tutorials for games and other projects in the 'Ideas' page!
- See if you can create a new project using the tutorials, or download another project that has been shared!