

technocamps



UNDEB EWROPEAIDD
EUROPEAN UNION



Llywodraeth Cymru
Welsh Government

Cronfa Gymdeithasol Ewrop
European Social Fund



Prifysgol
Abertawe
Swansea
University



CARDIFF
UNIVERSITY
PRIFYSGOL
CAERDYDD



PRIFYSGOL
BANGOR
UNIVERSITY



Cardiff
Metropolitan
University

Prifysgol
Metropolitan
Caerdydd

it.wales



PRIFYSGOL
ABERYSTWYTH
UNIVERSITY

PRIFYSGOL
Glyndŵr
Wrecsam

PRIFYSGOL
Wrexham
glyndŵr
UNIVERSITY

University of
South Wales
Prifysgol
De Cymru

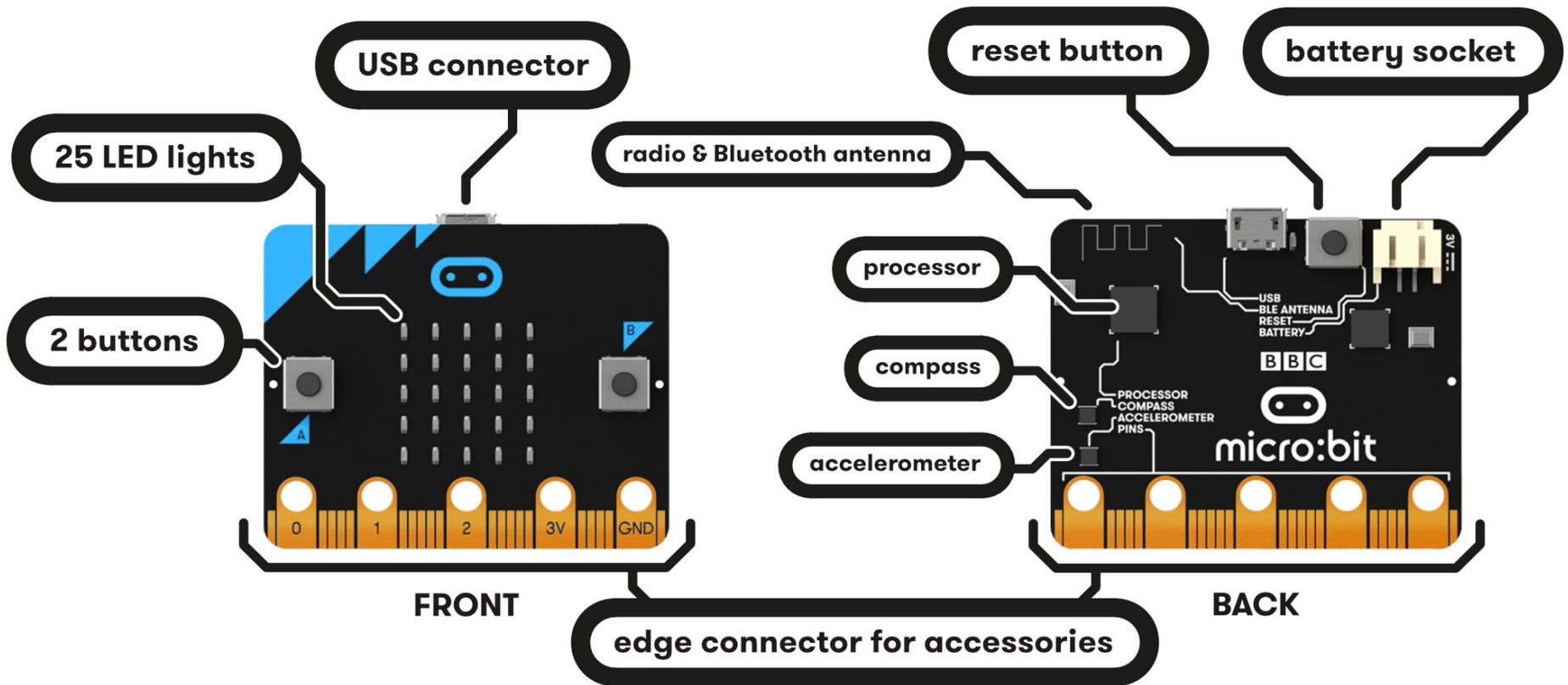
micro:bit Maths Game



Intro to micro:bit



What is a micro:bit?



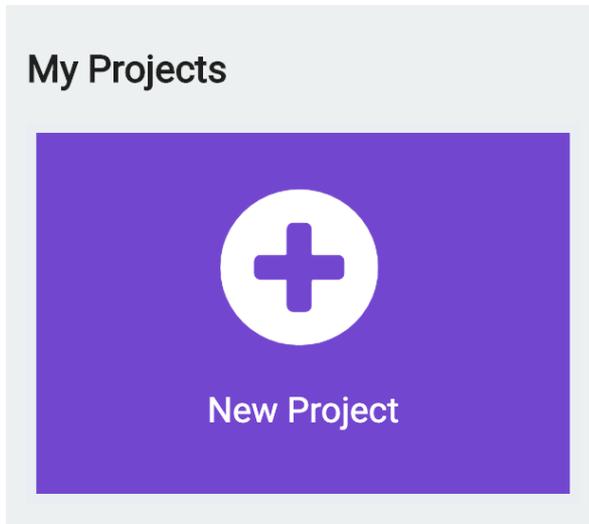
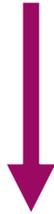
Starting with MakeCode

makecode.microbit.org

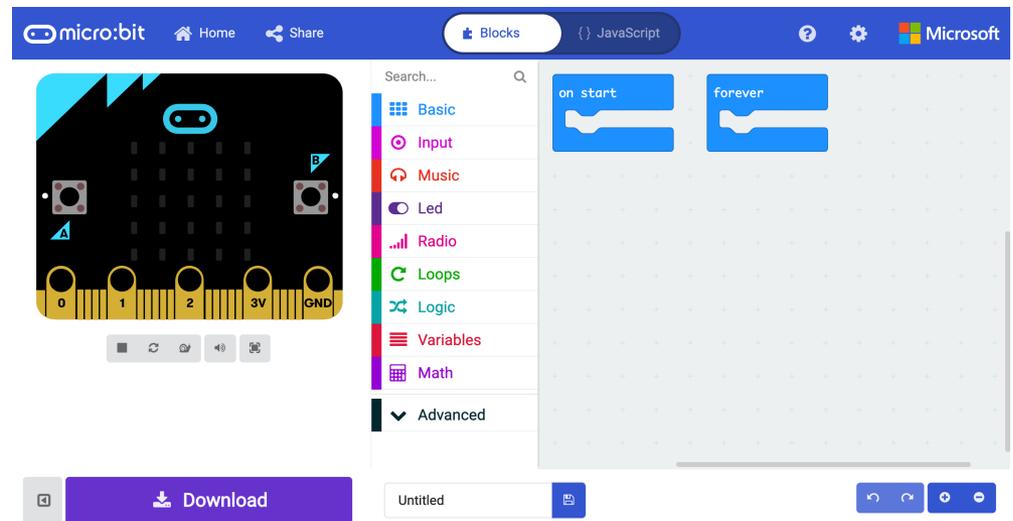
The screenshot shows the MakeCode website interface for micro:bit. At the top, there is a blue header bar with the 'micro:bit' logo on the left, a 'Home' button with a house icon in the center, and the Microsoft logo on the right. Below the header is a large green banner featuring a white circuit board with various components: a blue sensor, a black micro:bit board, a blue 'show leds' block, and a pink 'on shake' block. A white ribbon with colorful squares follows the path of the components. Below the banner is a 'My Projects' section with a purple 'New Project' button (a white plus sign in a circle) and two project cards: 'program1' and 'PelicanCrossing', each with a purple icon. An 'Import' button is located in the top right of the 'My Projects' section.

Starting with MakeCode

Click New Project

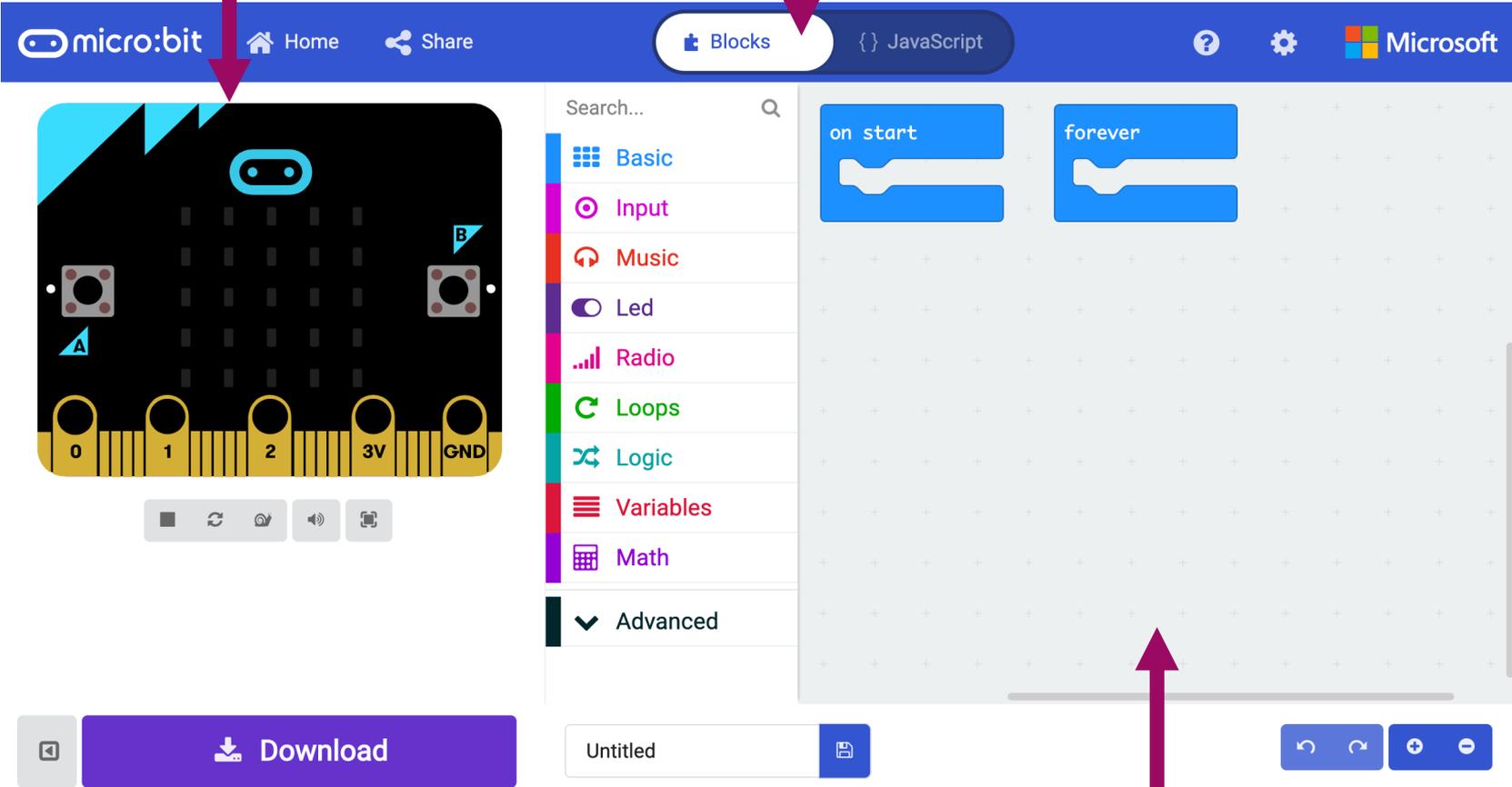


It should look like this!



Simulator

Language

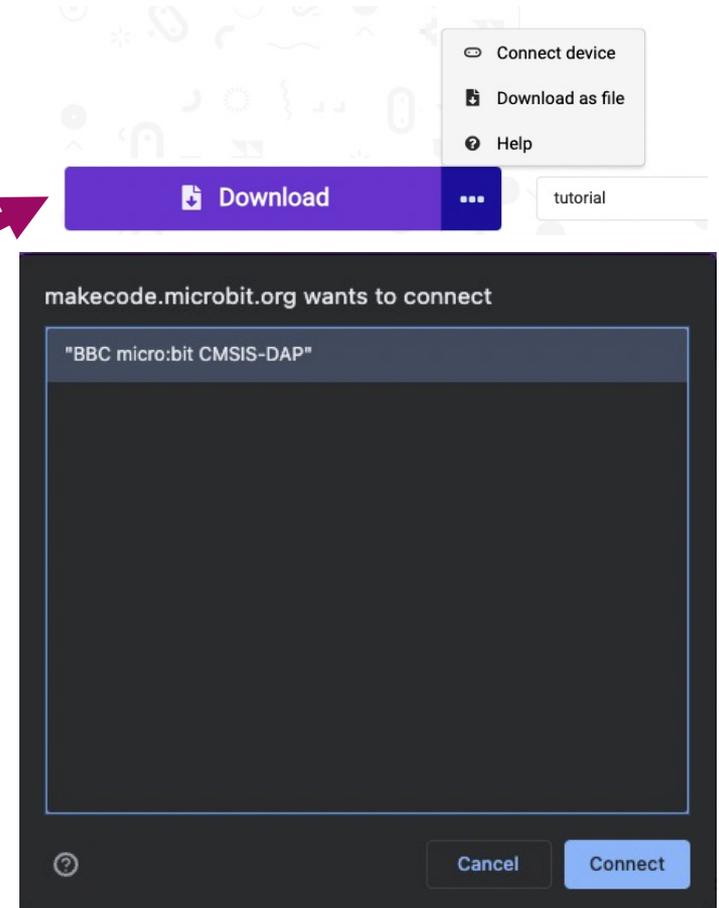


Save and Download

Code Window

Connecting the micro:bit

1. Plug the micro:bit into your computer
2. In the bottom left of your screen, click the 3 dots next to 'Download', then click 'Connect Device'
3. Follow the on-screen instructions until you see this popup
4. Click the name of your device (it should be the only option)
5. Click connect





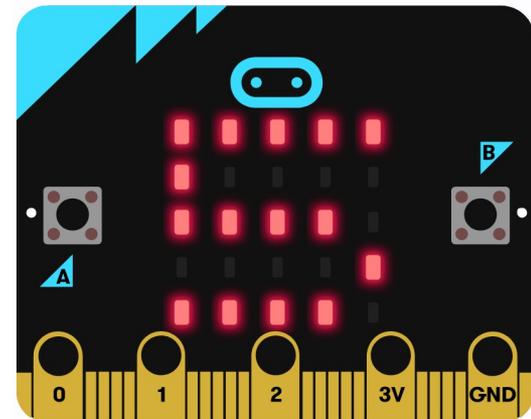
Activity: Numbers!

Numbers!

The micro:bit's screen can display any number.

1. Drag and drop the **'show string'** command from Basic into the **'forever'** block
2. Type any number into the **'show string'** block.
3. Click on **Download** at the bottom left.

What happens to the micro:bit?
Try choosing a different number!

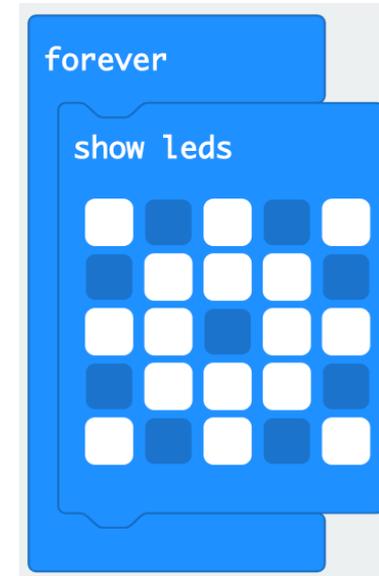
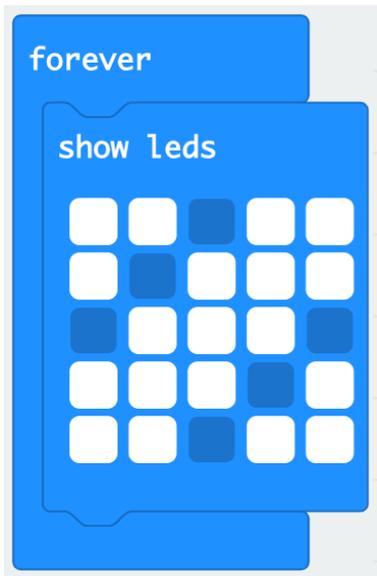


Custom Icons

The **'show leds'** command lets us choose exactly which LED lights we want to light up.

This can be used to create your own symbols or icons.

See which shapes you can make. Remember to click **Download!**

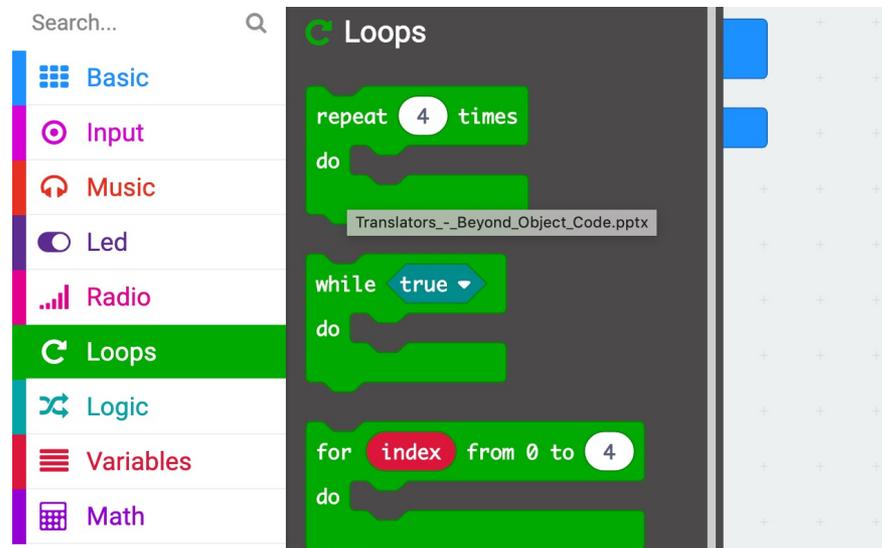




Activity: Changing Numbers

Loops

- Loops allow us to repeat commands.
- They can be repeated forever, for a certain number of times, or for a certain condition.
- These commands are found in the **Loops** section.



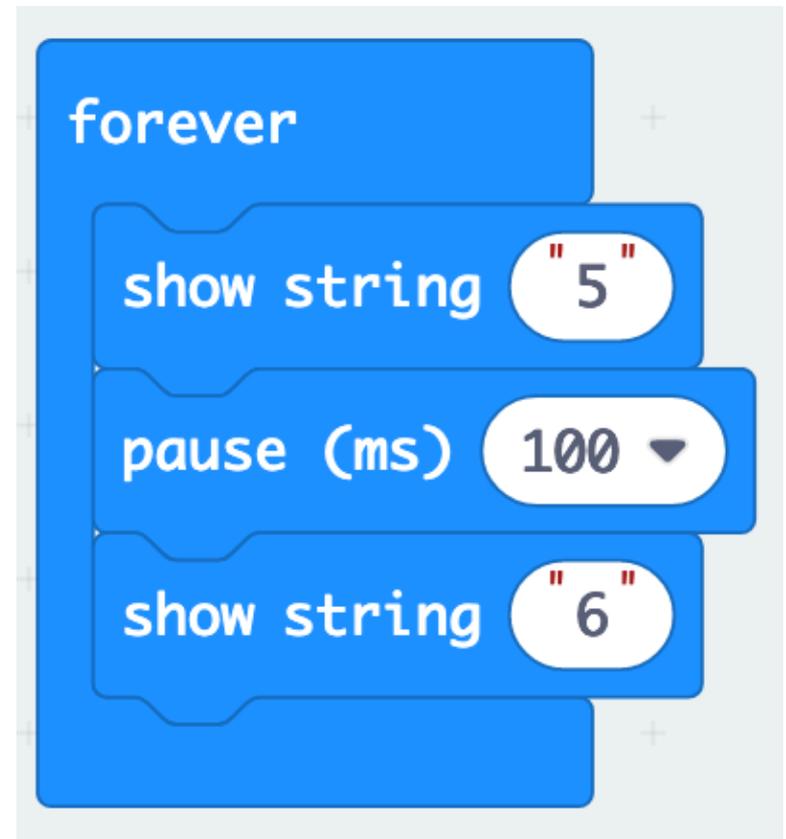
Default Loop

- We have a default loop command when we start the micro:bit project – the **'forever'** loop.
- The **'forever'** loop runs a set of commands until the micro:bit is unplugged or reset.
- You can only have one **'forever'** loop in the micro:bit code.



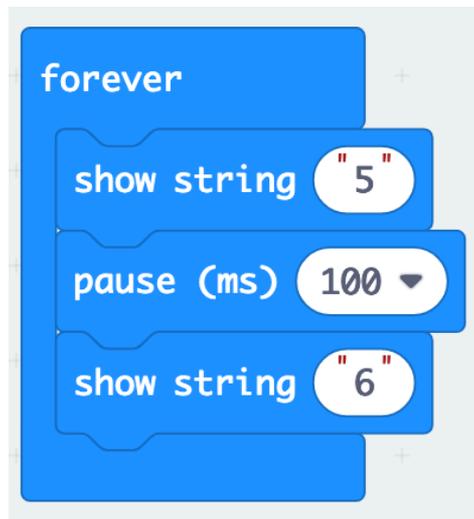
Changing Numbers

- We have created an application which shows a number on the micro:bit.
- Now we can extend our application to show the numbers counting.
- Add a **'pause (ms) 100'** command and add the next number after the pause.
- What happens when you download the program onto the micro:bit?



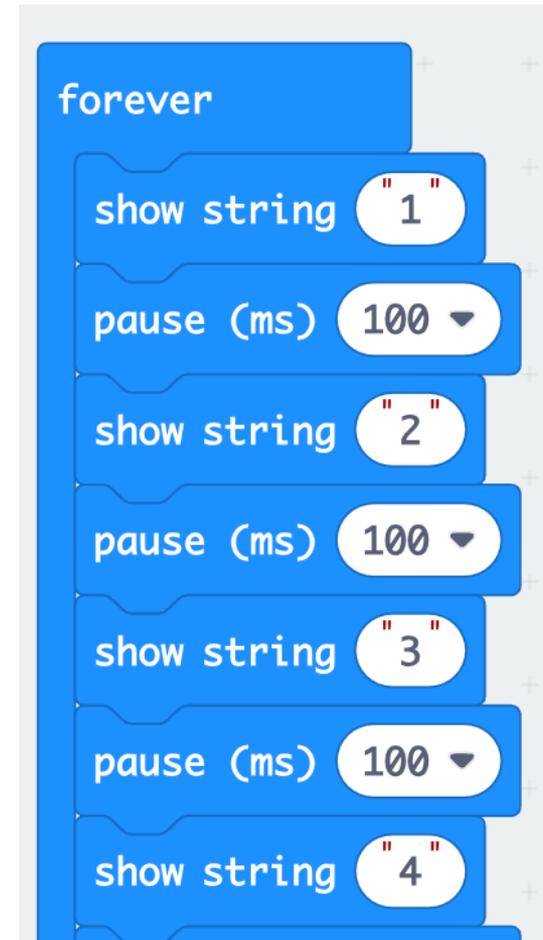
On Start vs Forever

- **'on start'** will run the code as soon as the micro:bit starts and ends once the code ends
- **'forever'** will run the code well... forever
- Try these two examples separately. What happens differently when you put the code in the **'on start'** loop instead of the **'forever'** block?



Counting with Forever Loops

- You can even use the **'forever'** loop to count
- Add **'show string'** blocks and more **'pause'** blocks until you can count from 1 to 9
- Because the code is in a **'forever'** loop, it will go back around to 1 after it reaches 9.
- What would happen if the code was in an **'on start'** block?

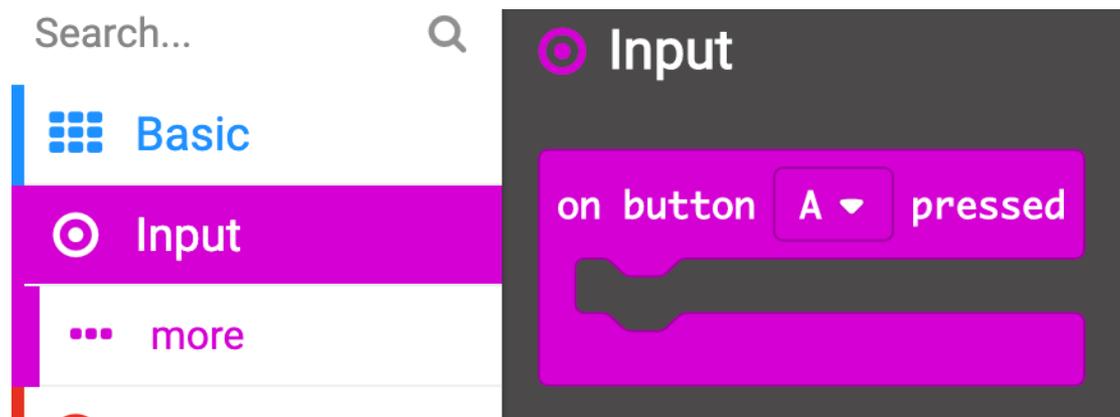


Events



Events

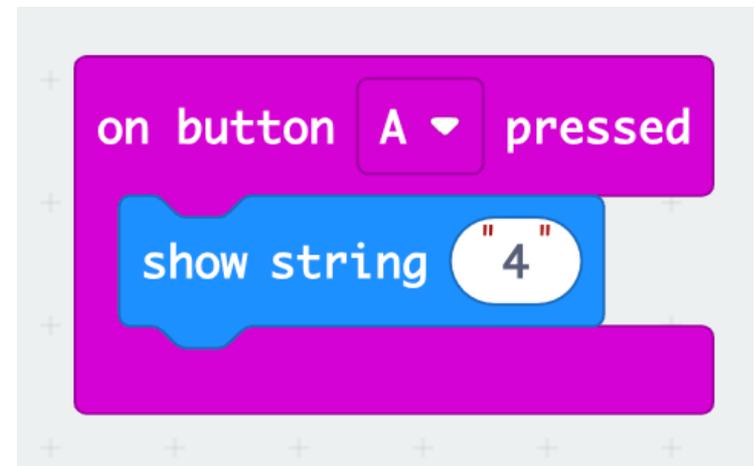
- Each micro:bit has two buttons, A and B.
- These buttons help us choose which action to take without reprogramming the micro:bit each time.
- For example, we can show one number when we press button A and another number when we press button B.
- The commands we need are found in the **Input** section.



Events

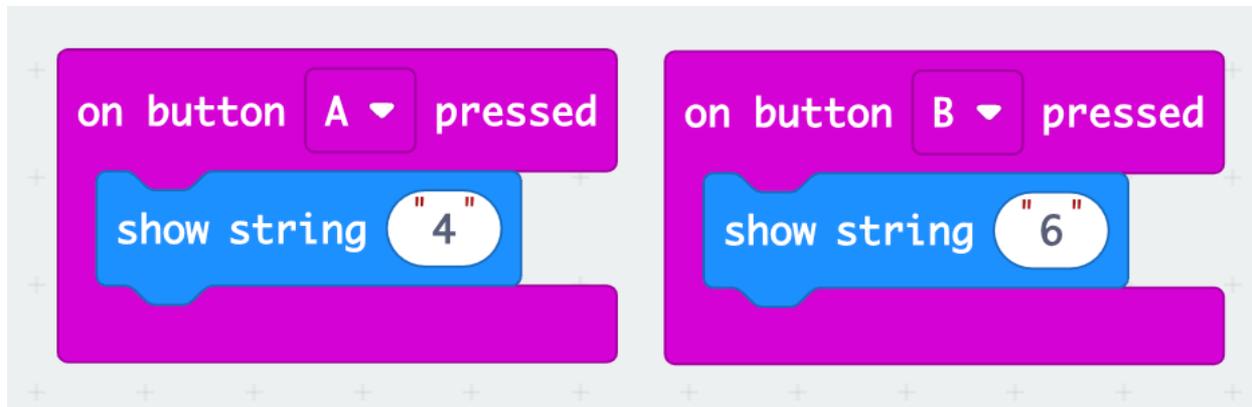
Let's try using an event for **button A**:

1. Clear your code.
2. Drag and drop the '**on button A pressed**' block from **Input** into your code
3. Now drag and drop the '**show string**' block from **Basic** into the **button A** block
4. Type in a number and **Download** the code. What happens?



Multiple Events

- We can use multiple **Input** commands – one for each button.
- Try adding an input for **button B**, the same way you did for **button A**
- Try running the code

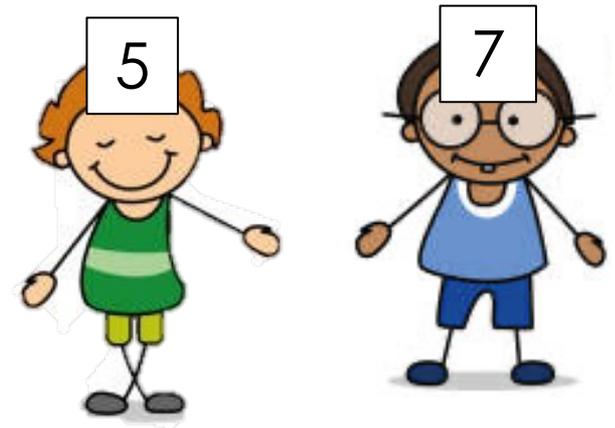


Activity: Salute!



How to Play Salute

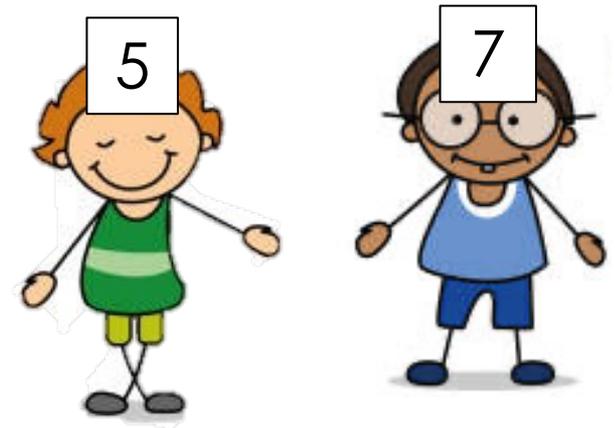
- Salute is a fun math game for 3 players.
- Two players hold up a number to their foreheads— they can see the other person's number, but not their own!
- A third player looks at the two numbers and decides either to multiply them or add them. They say the result out loud.
- Based on that number and the other person's number, the two other players try to guess their own number.
- The fastest player to guess their own number wins the round!



How to Play Salute

For example:

- The green shirt player sees the number 7 on the blue player
- After hearing the answer is 12, the player knows that no number can multiply with 7 to equal 12.
- Therefore he knows the numbers must have been added.
- What adds with 7 to equal 12?
- 5!



Playing Salute with micro:bits

We can play Salute with micro:bits. micro:bits have a way to generate random numbers, and there is even an event to detect when the micro:bit is held up.

Let's try it now!

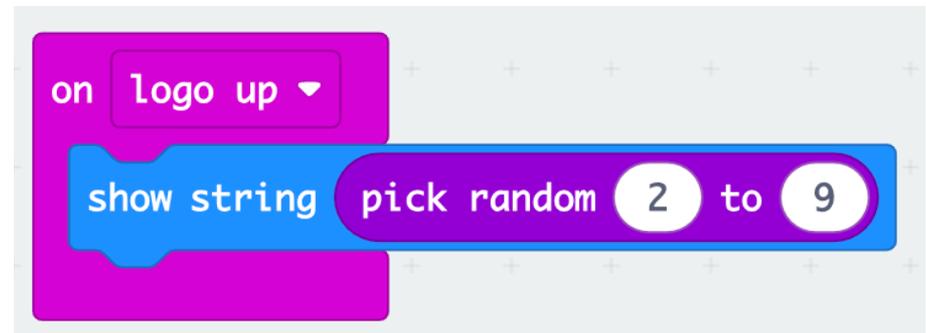
1. Clear your code
2. Drag and drop the **'on shake'** block from **Input**
3. Change the **'shake'** event to **'logo up'**



Playing Salute with micro:bits

4. Drag and drop the 'show string' block from **Basic** into the 'on logo up' block.
5. Choose the 'pick random ...' block from **Math** and drag it into the 'show string' block.
6. Set the numbers to **1** to **9** in the 'pick random ...'. For an added challenge, add bigger numbers.

Try holding the micro:bit up to your head. A random number will appear. Time to play!





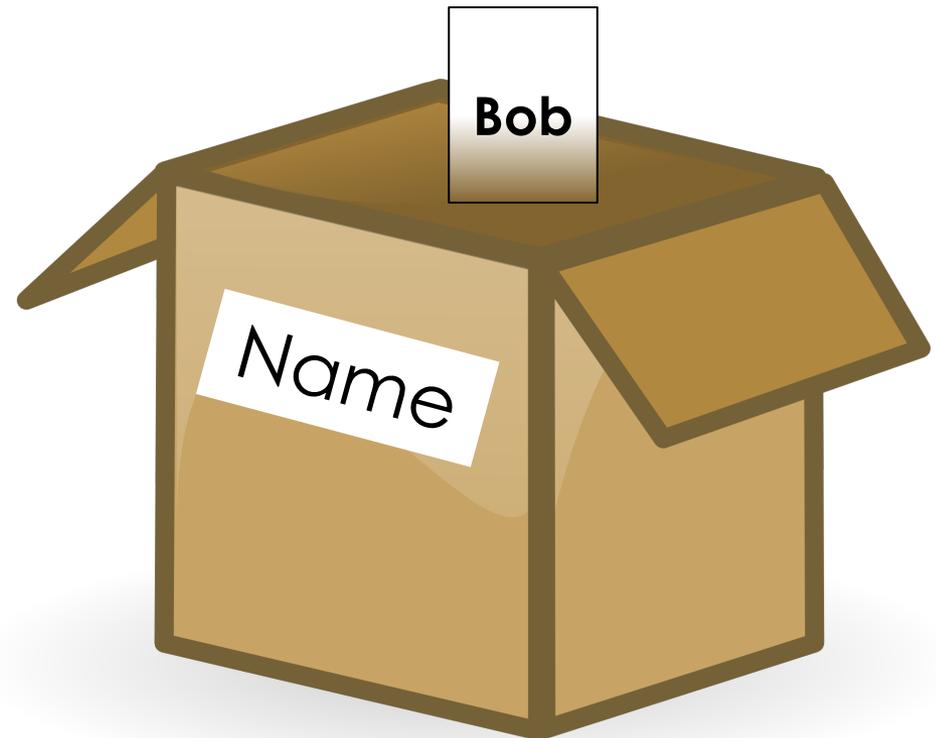
Variables

What Is a Variable?

A **variable** is something that stores data in our program. It is like a box with a label on it.

I can store different things in the box, but the label stays the same.

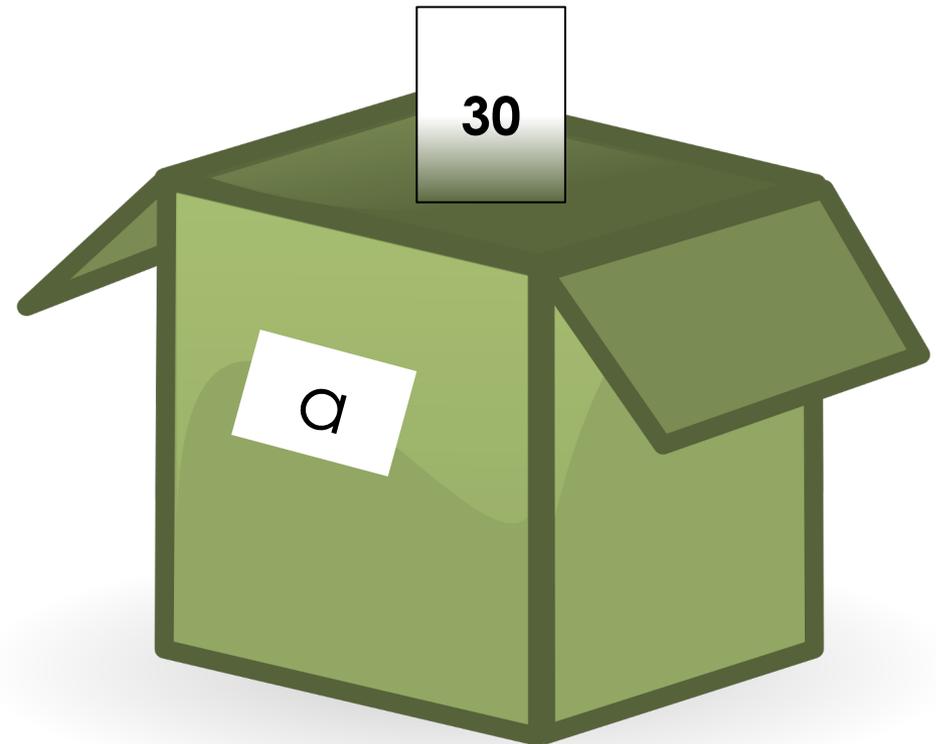
For example, I have stored the word “Bob” in my **variable** which is labelled “Name”.



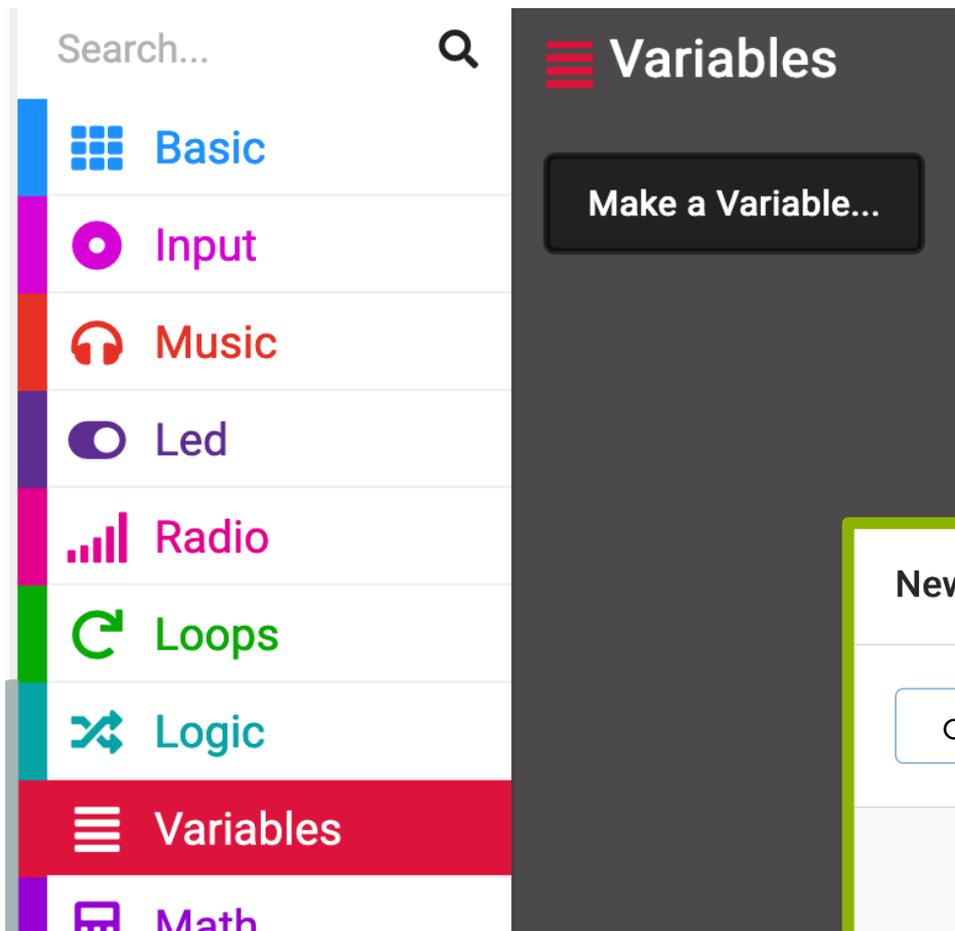
Humidity Variable

We will make a **variable** named 'a' which will remember the most recent reading from the sensor.

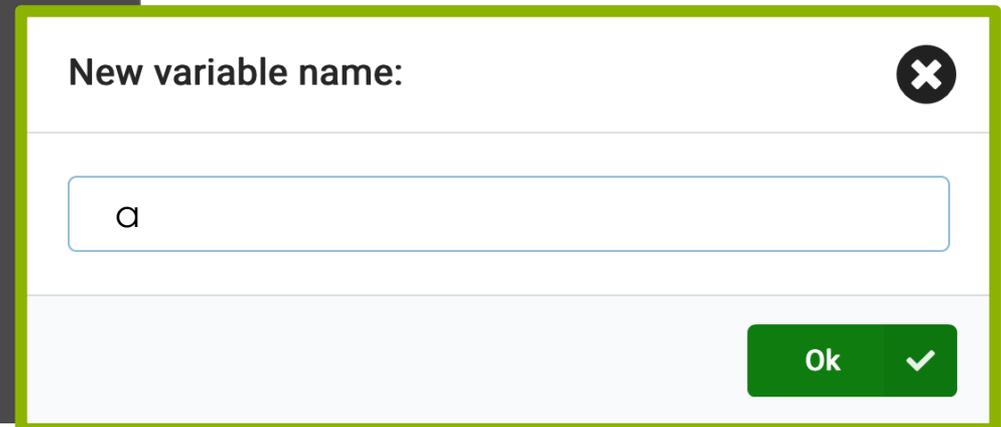
If we program our micro:bits correctly, this will help our program to work properly, even if the sensor gives an error message.



Fixing the Sensor

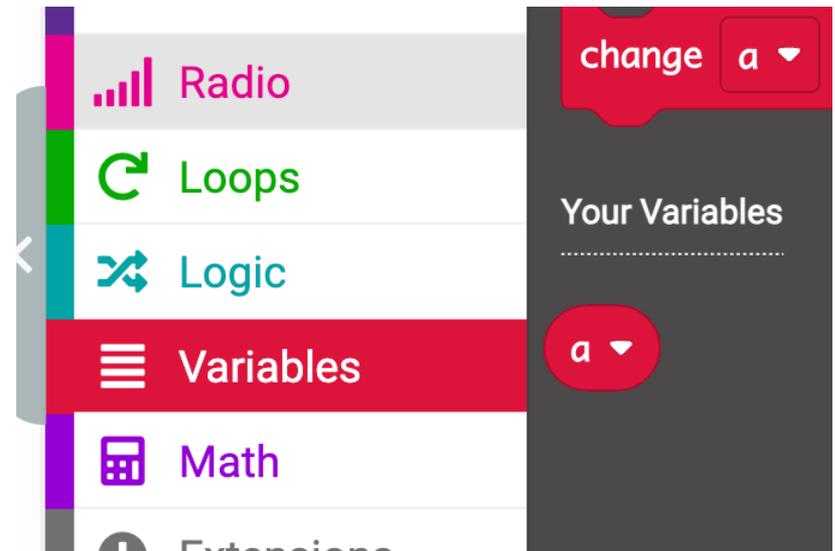


- Click the red **Variables** tab
- Click '**Make a Variable...**'
- Enter the name as 'a'



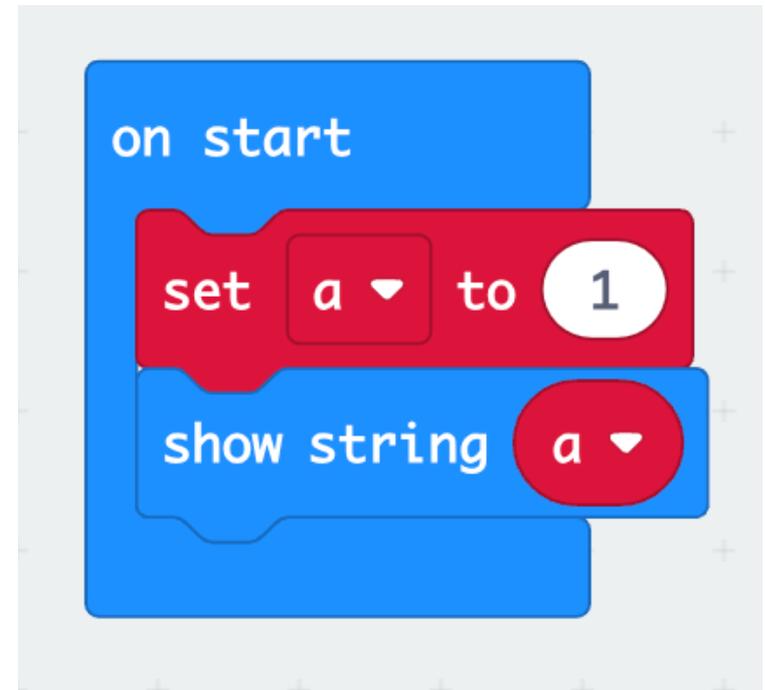
Variables

- In micro:bit variables are found under the **Variables** section.
- We can use variables to do all sorts of things. Remember the counter we made earlier? We can make one that we can control ourselves using variables.



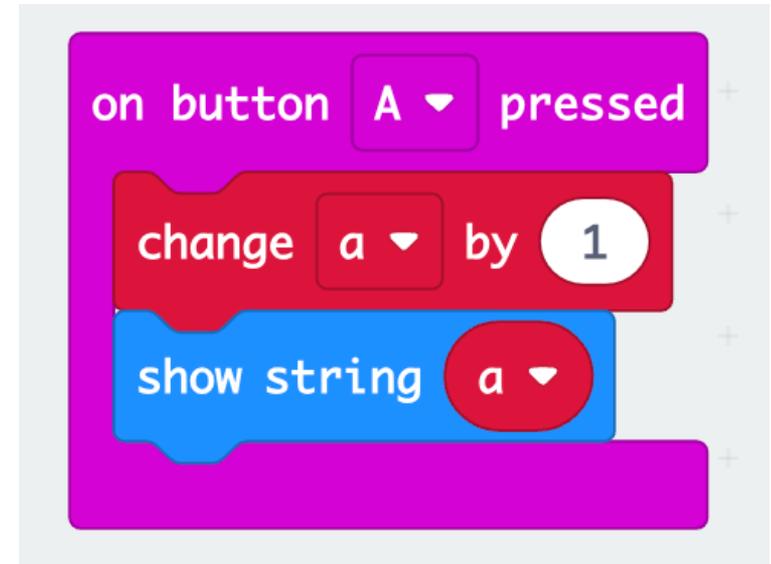
Making a Counter

1. Add an **'on start'** block
2. Go into the **Variables** section and choose the **'set a to 1'** block
3. Add a **'show string'** block and go back to **Variables**
4. Drag and drop the **'a'** from 'Variables' into the **'show string'** block



Making a Counter

5. Add an **'on button A pressed'** event.
6. Go back to **Variables** and choose the **'change a by 1'** block.
7. Copy the **'show string a'** block from before and add that too.



Making a Counter

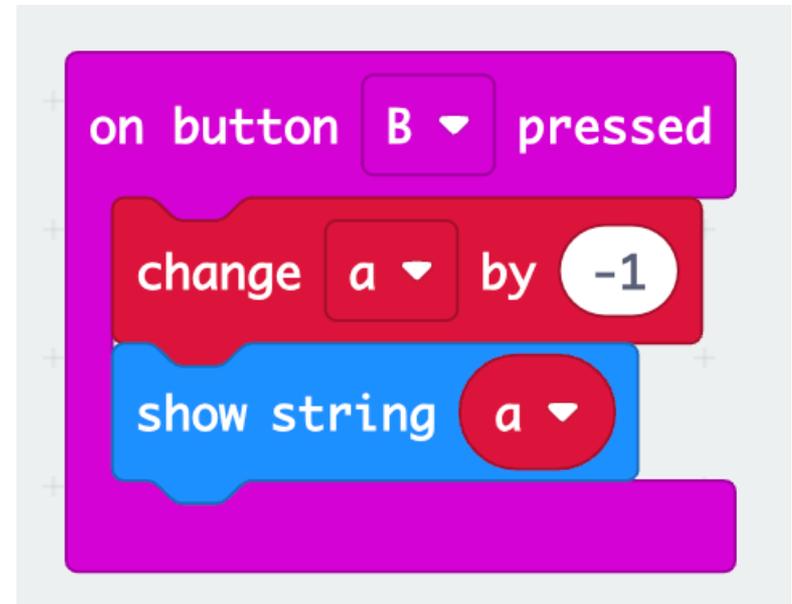
It's even possible to decrease the counter using **Button B**. Can you figure out how to do it?

Hint: you can use negative numbers in the '**change a by ...**' block.

Making a Counter

It's even possible to decrease the counter using **Button B**. Can you figure out how to do it?

Hint: you can use negative numbers in the **'change a by ...'** block.



Activity: Scoreboard



Making a Scoreboard

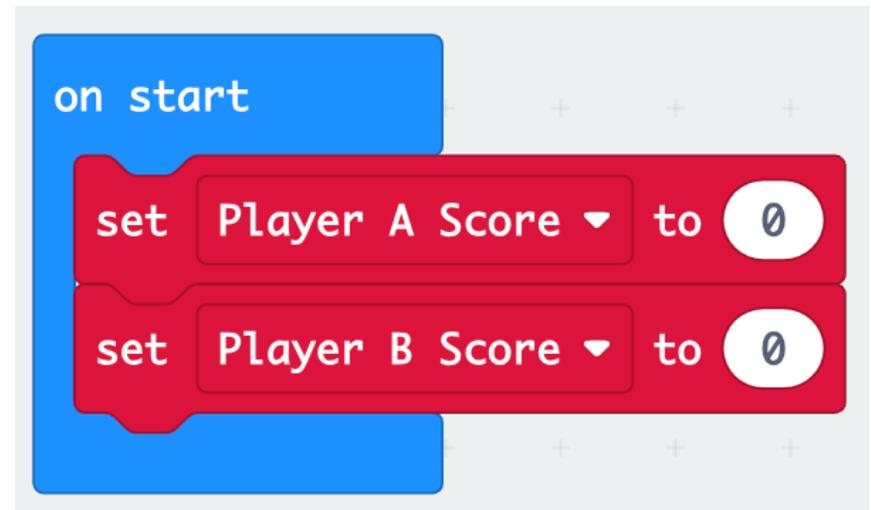
Now we'll make a scoreboard for our game of Salute! The third player will keep track of the score.

1. You will need to create two **variables** - one for player A and one for player B
2. Set Player A Score and Player B Score to 0 using the '**on start**' block.

Making a Scoreboard

Now we'll make a scoreboard for our game of Salute! The third player will keep track of the score.

1. You will need to create two **variables** - one for player A and one for player B
2. Set Player A Score and Player B Score to 0 using the '**on start**' block.

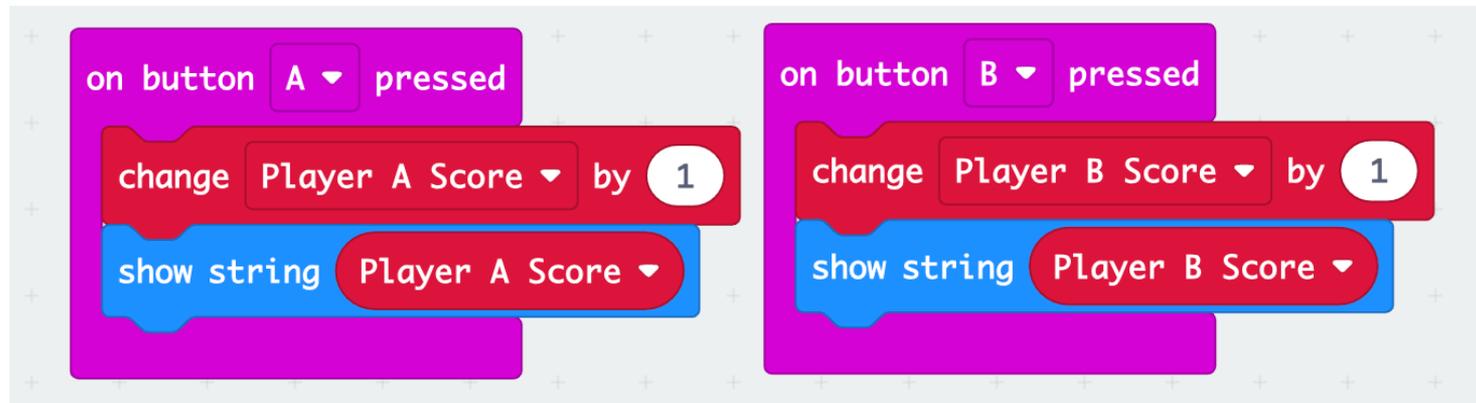


Making a Scoreboard

3. Set **button A** to increase **Player A Score** by 1 and display it.
4. Set **button B** to increase **Player B Score** by 1 and display it.

Making a Scoreboard

3. Set **button A** to increase **Player A Score** by 1 and display it.
4. Set **button B** to increase **Player B Score** by 1 and display it.



Making a Scoreboard

Once we increase the score, we want to clear the screen.

5. Under each 'show string' block, add a 'pause (ms) 1000' block followed by a 'clear screen' block from Basic.

The image shows two Scratch code snippets side-by-side. The left snippet is for 'on button A pressed' and contains four blocks: 'change Player A Score by 1', 'show string Player A Score', 'pause (ms) 1000', and 'clear screen'. The right snippet is for 'on button B pressed' and contains four blocks: 'change Player B Score by 1', 'show string Player B Score', 'pause (ms) 1000', and 'clear screen'. Both snippets are enclosed in purple containers.

```
on button A pressed
  change Player A Score by 1
  show string Player A Score
  pause (ms) 1000
  clear screen

on button B pressed
  change Player B Score by 1
  show string Player B Score
  pause (ms) 1000
  clear screen
```

Making a Scoreboard

Now we need a way to check the current scores. There is an event for when both Button A and B are pressed at the same time.

1. Add an **'on button A+B pressed'** block
2. Now, add a **'show string'** block to show **'Player A Score'**.
3. Add a **'pause (ms)'** block and set it to **1 second**.
4. Add another **'show string'** to show **'Player B score'**.
5. Add another **'pause'** block and clear the screen.

