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

i.t.wales

PRIFYSGOL ABERYSTWYTH UNIVERSITY

PRIFYSGOL glyndŵr Wrecsam
Wrexham glyndŵr UNIVERSITY

University of South Wales
Prifysgol De Cymru

technocamps

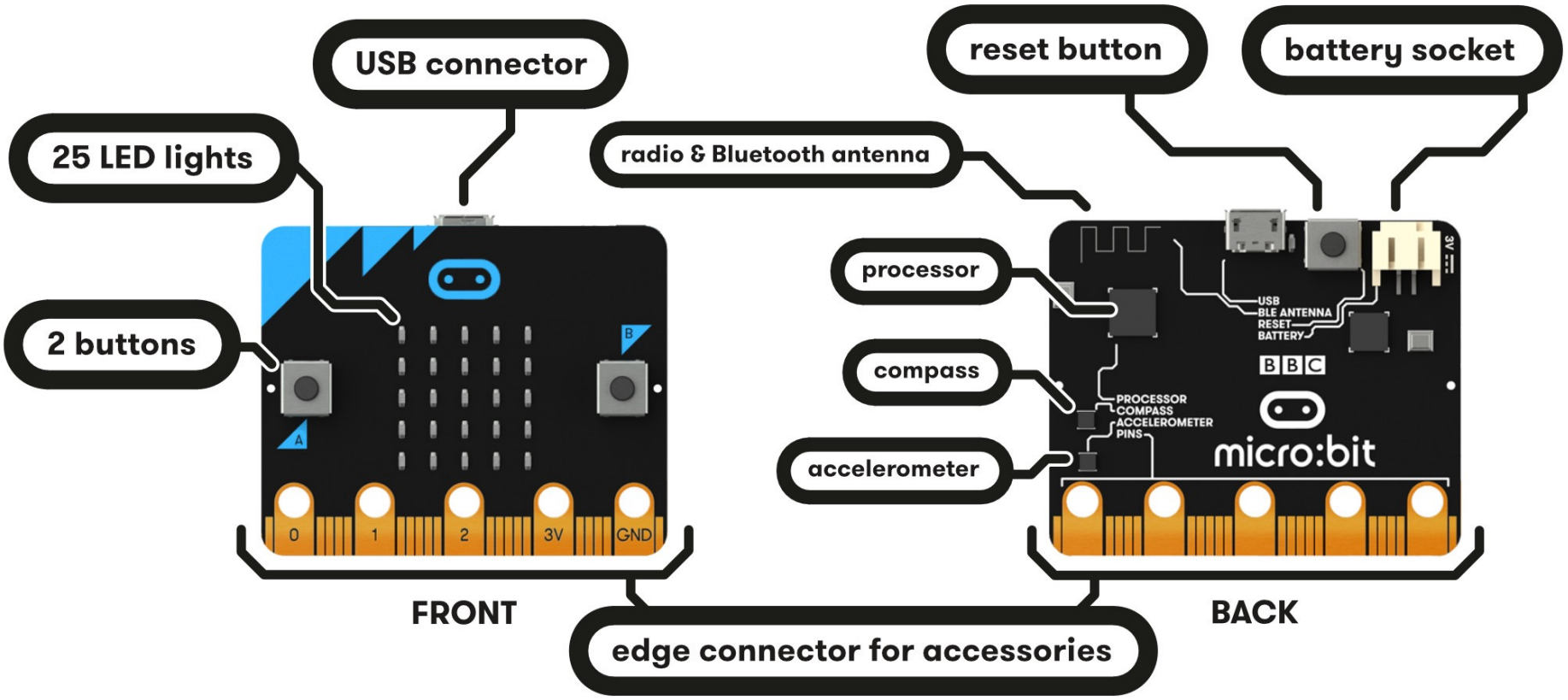
Musical micro:bit

technocamps

Intro to micro:bit

technocamps

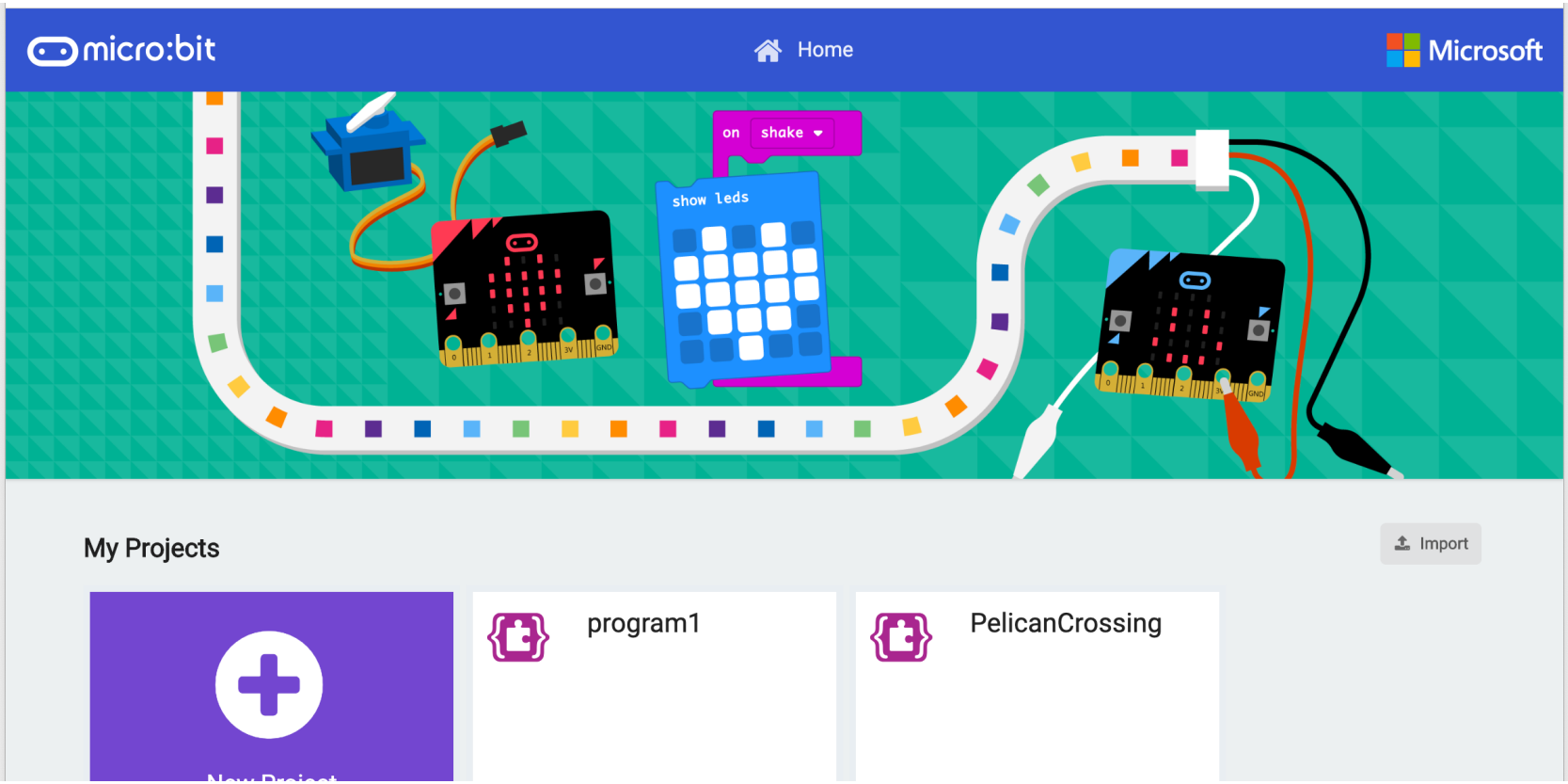
What is a micro:bit?



technocamps

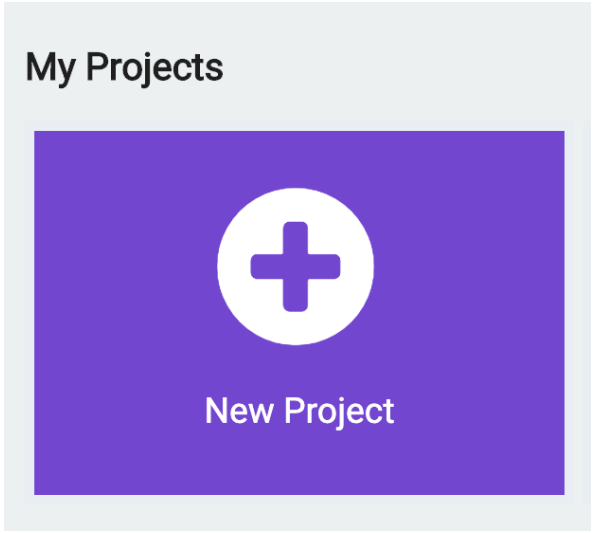
Starting with MakeCode

makecode.microbit.org

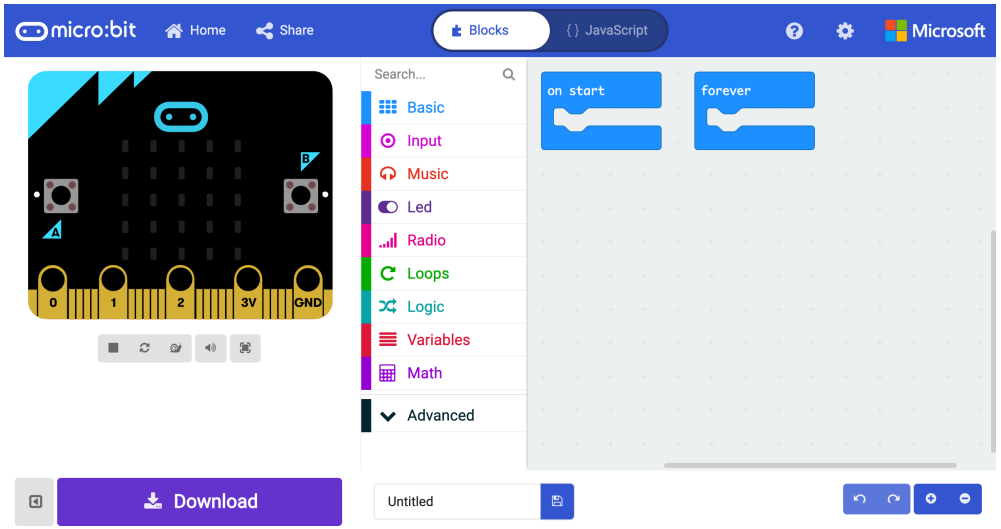


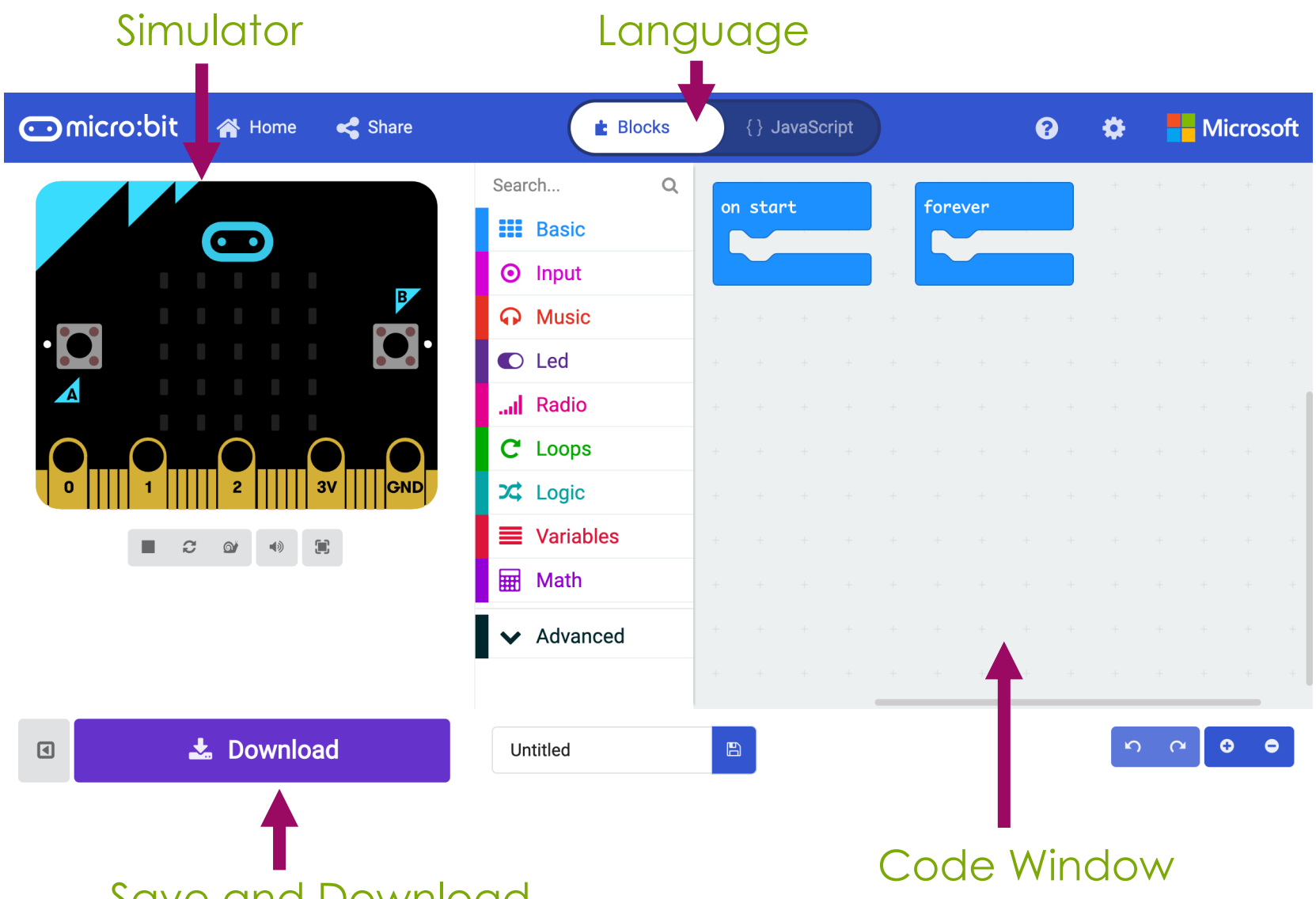
Starting with MakeCode

Click New Project



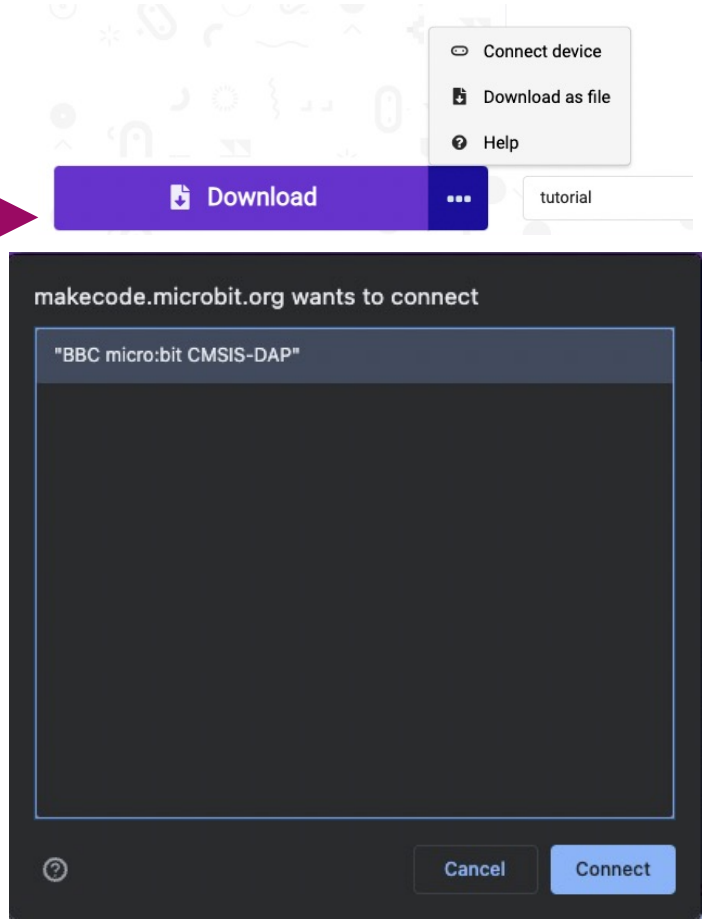
It should look like this!




Simulator
Language
micro:bit
Home
Share
Blocks
{ } JavaScript
Microsoft
Search...
Basic
Input
Music
Led
Radio
Loops
Logic
Variables
Math
Advanced
on start
forever
0
1
2
3V
GND
Download
Untitled
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: Melody

A Single Note

Let's start by programming the micro:bit to play a single note:

1. Click on the Music section.
2. Click on the play tone Middle C for 1 beat command.
3. Drag and drop it into the on start block.
4. Click on the Middle C and choose a note to play.
5. Click on download.

What happens to the micro:bit?

Try changing the note.

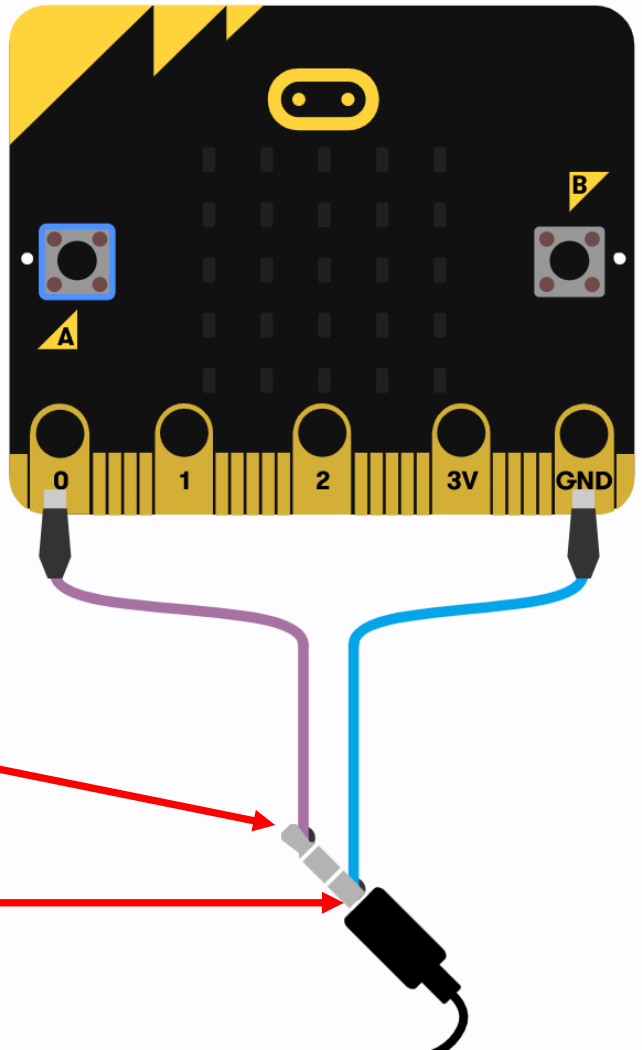
Melody

- MakeCode also has preset melodies that we can use.
- Try playing one of these melodies by adding the start melody dadadum repeating once command to the on start block.
- Try a few different melodies!

Activity: Headphones

Connecting headphones

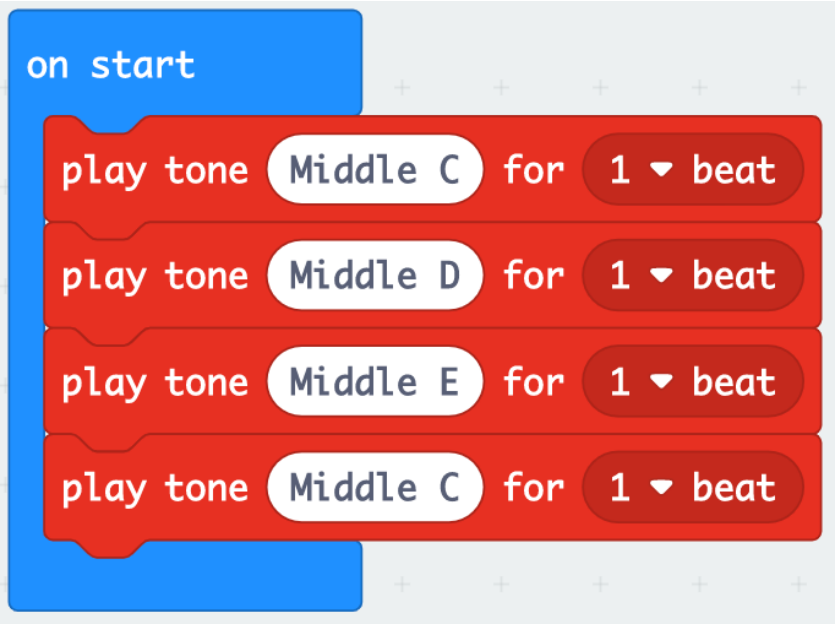
- With so many micro:bits playing music at the same time, it can be difficult to hear your own.
- We can connect headphones by attaching them to the pins at the bottom of the micro:bit with crocodile clips.



- Connect pin 0 to the tip of the headphone plug.
- Connect the GND pin to the longer part of the headphone plug

Make Your Own Music

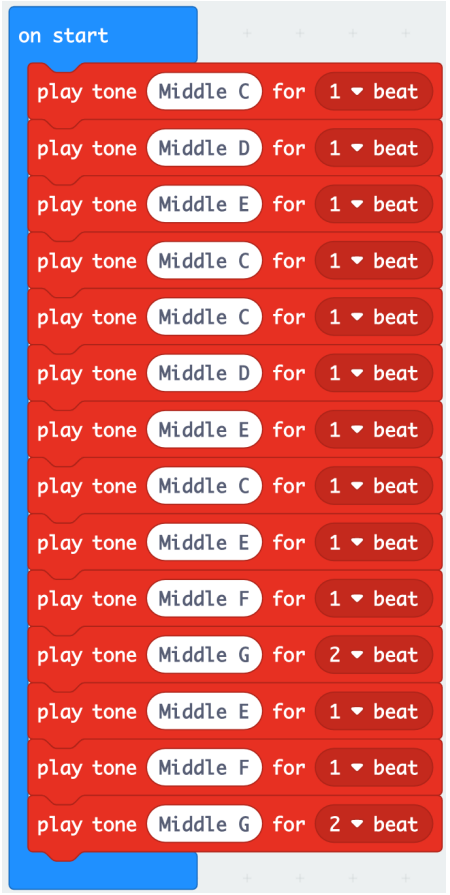
- We've learnt how to get our micro:bit to play a single note and a preset melody.
- Now we can try and create our own melody.
- Add several play tone commands, one after the other, in the on start block.
- Be creative and see what melodies you can make!



Activity: On Repeat

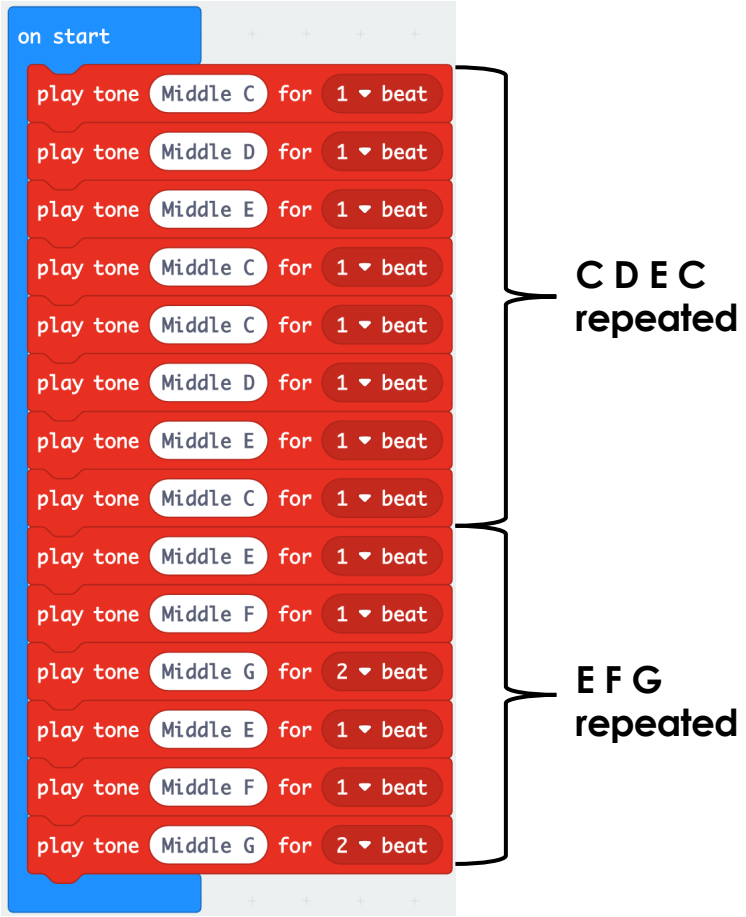
Extended Melody

- What if we wanted to play a much longer melody?
- Try building and running this code.
- The code only plays a short tune but includes a lot of blocks.
- Now imagine how many blocks we would need if we wanted to play a song that was twice as long, or 5 times, or 100 times!
- How could we do this without adding individual blocks for each note?



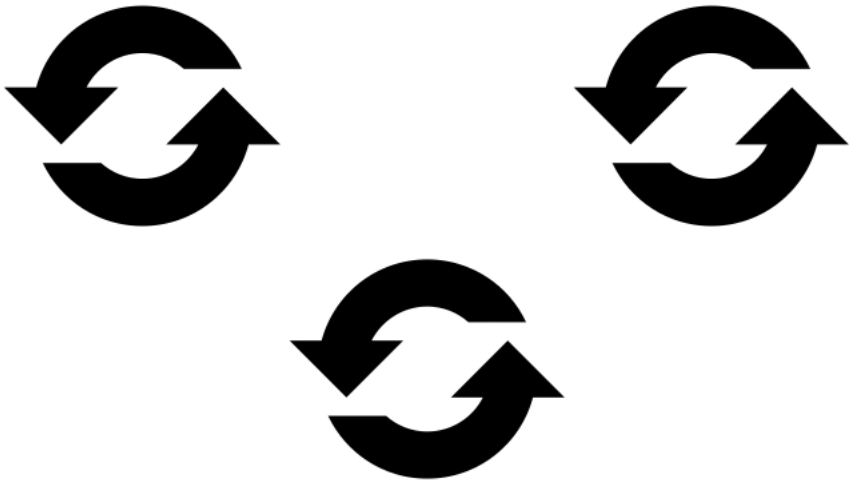
Extended Melody

- Our song has repeated blocks of notes.
- Instead of adding new blocks every time, we can use loops to repeat these segments.



Loops

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



Default Loop

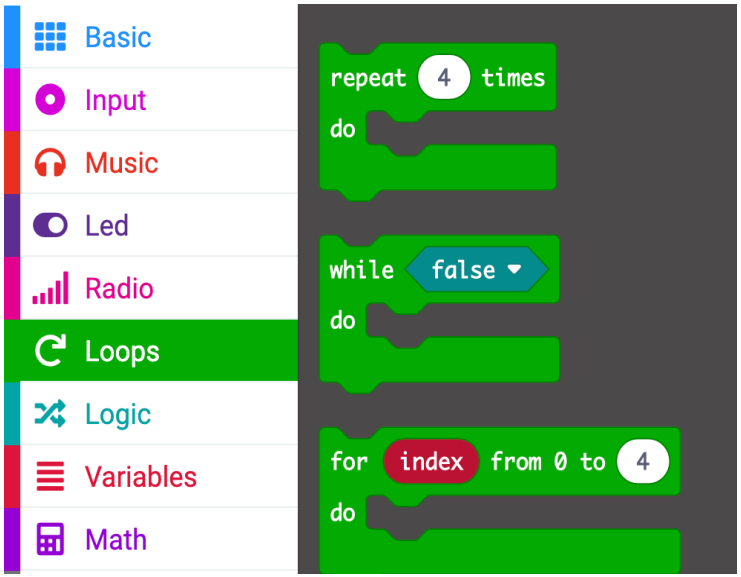
- We have a default loop command when we start the micro:bit project – the forever loop.
- The forever loop runs a set of commands until the micro:bit is unplugged or reset.
- You can only have one forever loop in the micro:bit code.

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 dragging your melody to the forever block. What happens differently when you put the code in the on start loop instead of the forever block?

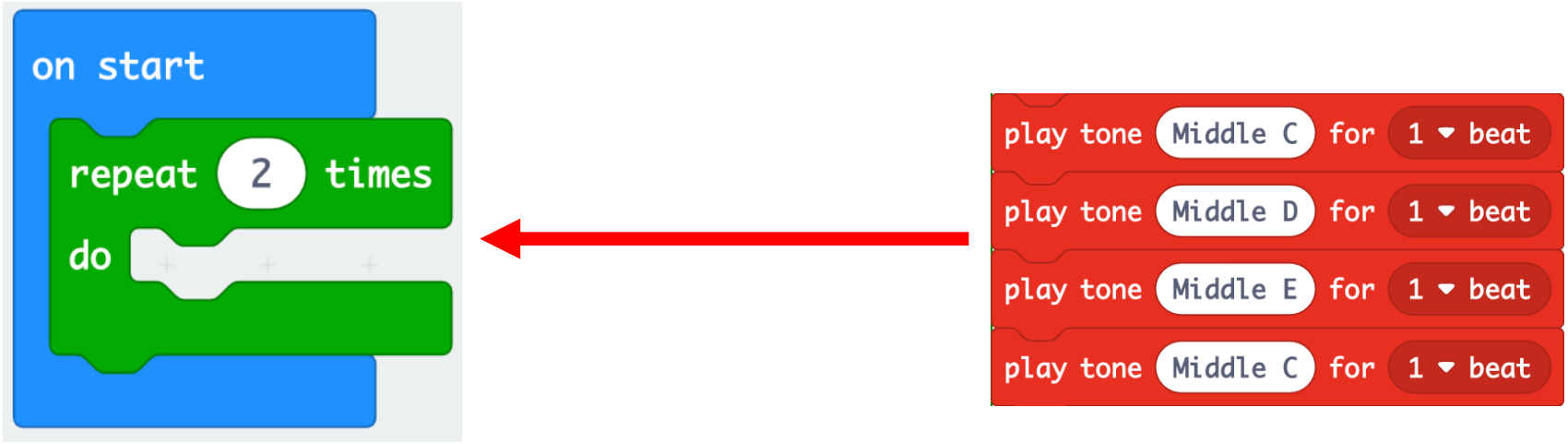
Specific Loops

- We can also have more specific loops that don't necessarily repeat forever.
- For instance, we can use the repeat x times block to repeat a code segment a certain number of times.
- For our melodies this is useful if only certain parts of the melody need to be repeated. Then we don't need to use a play tone command every time.
- Let's try to make a tune!



Melody Loop

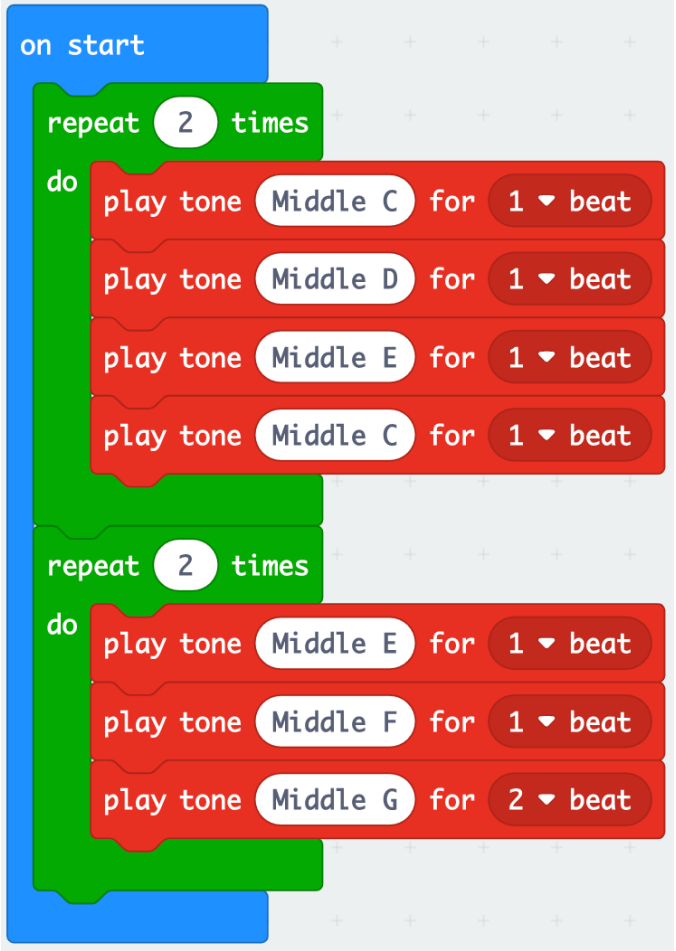
- Begin by drag and dropping the repeat x times block into the on start block.
- Add the play tone commands to the new loop.

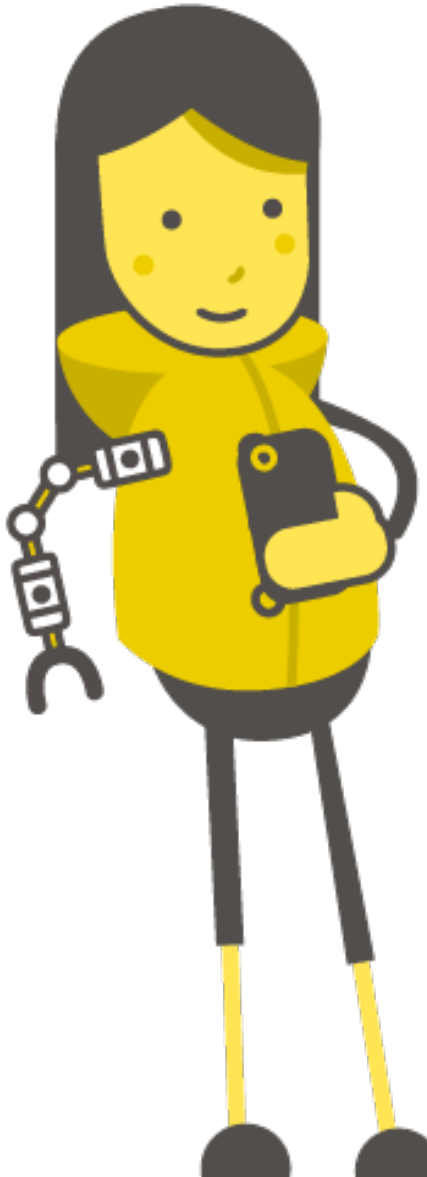


Melody Loop

- Add another repeat loop with some more play tone commands.
- Try running the code!

- Now the code is cleaner and more efficient. If we want to repeat it more times, we can just change the number on the loop.





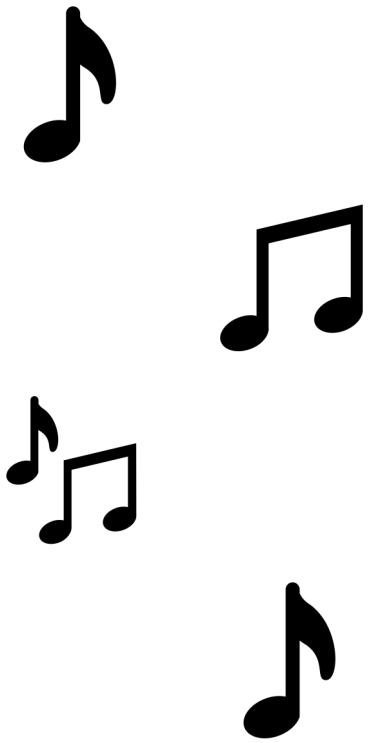
Activity: Jukebox

Jukebox

- A jukebox is a music playing device that gives you the choice of multiple songs to play.

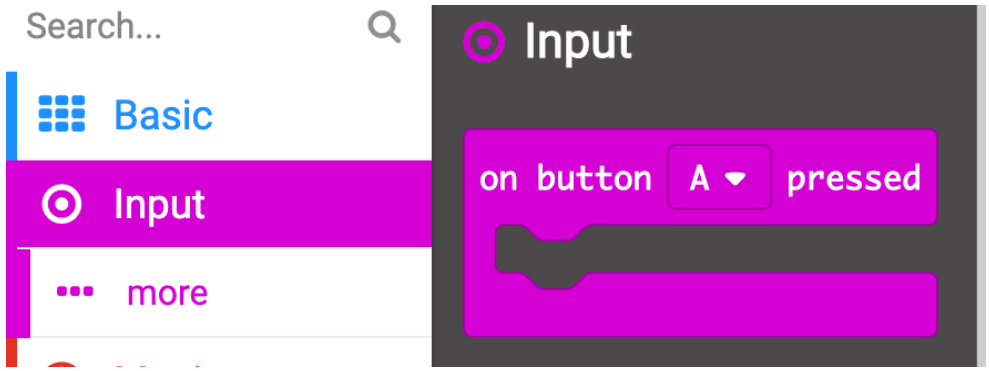
- So far, our micro:bits can only play one song. How can we program it to play several different songs? And how can we allow the user to choose which one to play?

- We need to use 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 play one song when we press button A, and a different song 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. Click on Input
2. Drag and drop the on button A pressed block into your code
3. Now click on basic and drag and drop the play tone command into the event block
4. Choose you're a note to play 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.

Jukebox

- Create your own jukebox by adding a different song to each input block.

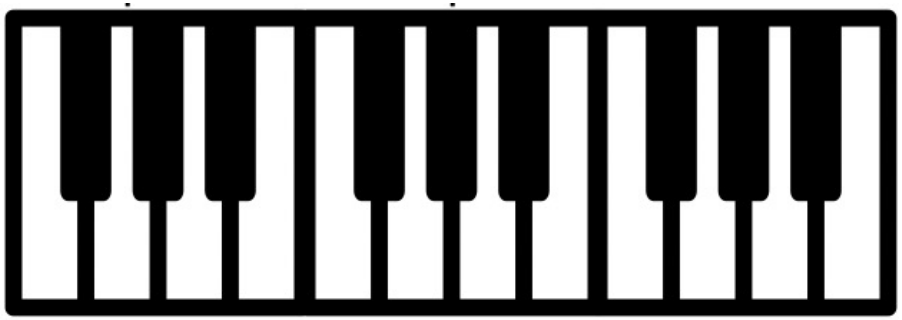
- You can make your own melody using the play note commands or use the existing melodies in the Music section.

technocamps

micro:bit Piano

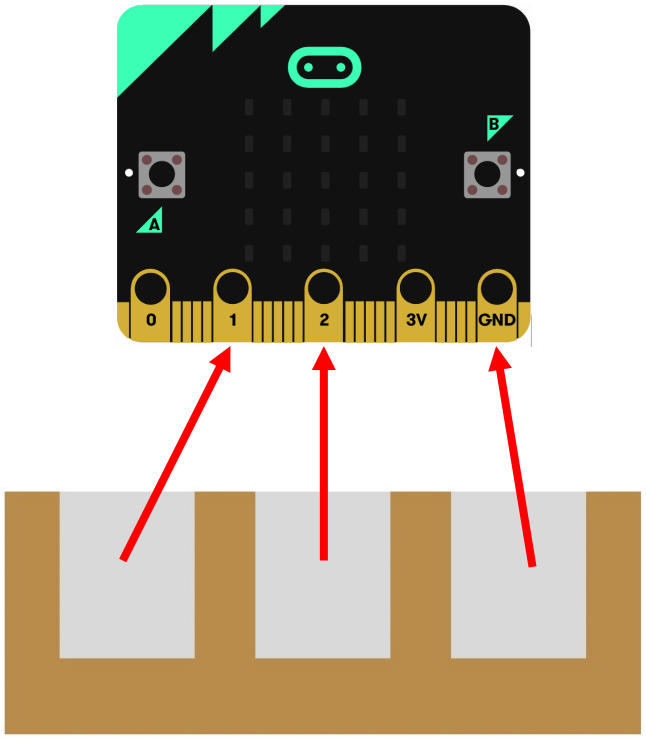
Instrument Input

- Instead of using the buttons to play notes or a melody, we can connect our own input device.
- To do this, we use the pins at the bottom of the micro:bit, just like you did to connect the headphones.
- We can make our own instrument. When we play it, an input signal will be sent to the micro:bit. The micro:bit will then produce output.



Connecting the Piano

- Attach three crocodile clips to the cardboard piano.
- Connect the other ends of the crocodile clips to pin 1, pin 2, and the GND pin.



Activity: Piano Code

Pin Input

- Instead of using the buttons as input, we use the pins. The "keys" on our piano are connected to the pins.
- When we press on one of our keys and the ground pin, we complete an electrical circuit.
- Try adding input for pin P1 and pin P2. Give the code some music to play, run the code, and see what happens when you play your piano.

Pin Input

This is what your code should look like:

Experiment with some different musical outputs and see which songs you can play.

Activity: Chords

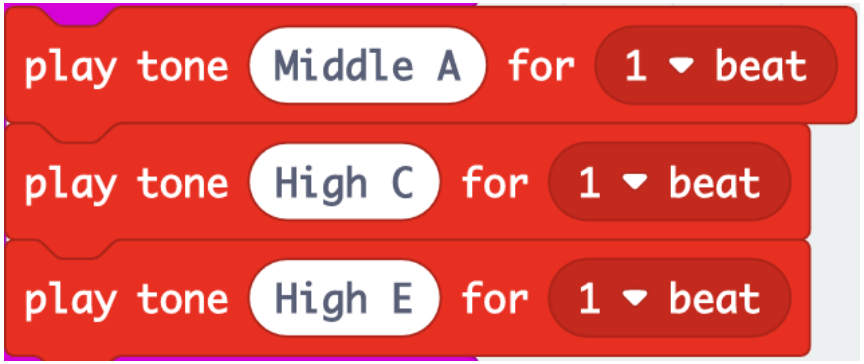
Chords

- Chords are a combination of multiple notes playing at the same time, to make a sound.
- The micro:bit can't really play a chord since it can only play one note at the time.
- However, if we play three notes quickly, one after the other, it sounds like a "broken" chord.

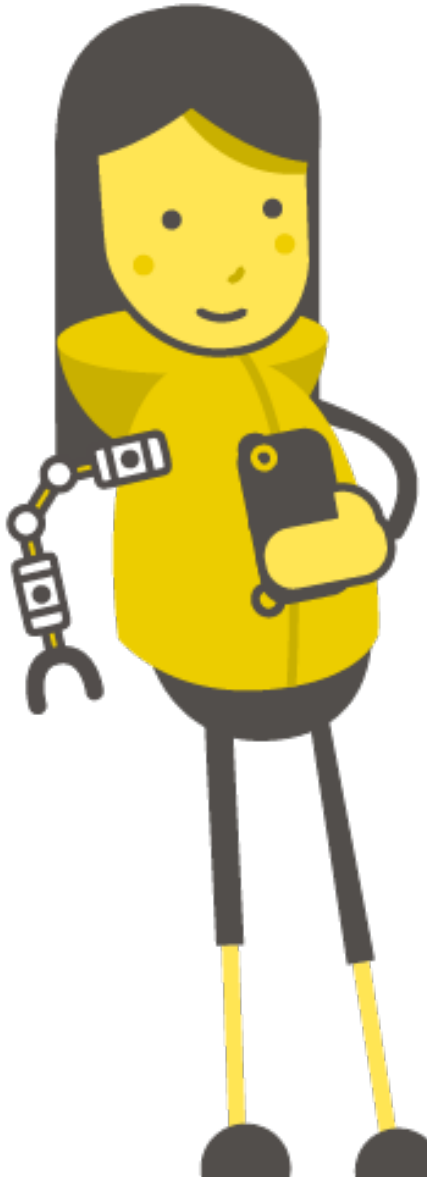
Chords

- These two sets of notes make chords. Try adding them to your code and see what it sounds like.

F Major



A Minor



Activity: Octaves

Changing Pitch

- Our pianos can now play notes or chords, but we can't change these sounds without reprogramming and downloading the code.
- We can add more input blocks to higher or lower the pitch of the note/chord without changing the code every time. The pitch of a note doubles when you move up an octave, and halves when you move down an octave.

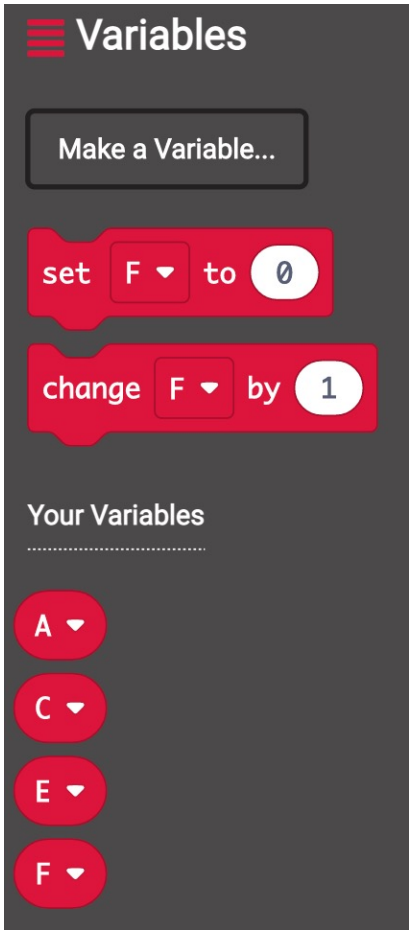
- To do this, we need to define the notes as variables that can be changed.

Variables

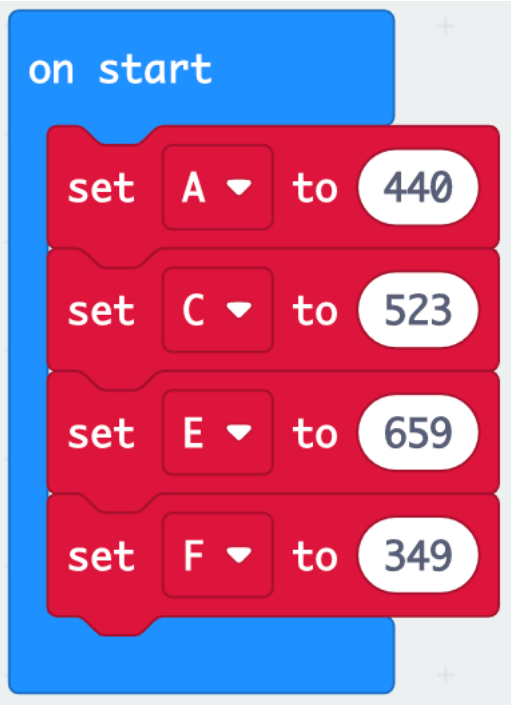
- Variables are items that can be remembered and changed by the micro:bit.
- They can take different forms such as a number or a text.
- We can change their value and use them in many ways in our code, but first we need to define them.
- They are found in the Variables section on MakeCode.

Creating Variables

- To create a variable, click on the Variables section and "Make a Variable".
- Make four variables: A, C, E, and F representing four musical notes.
- There should now be a few commands available to use in the Variables section.



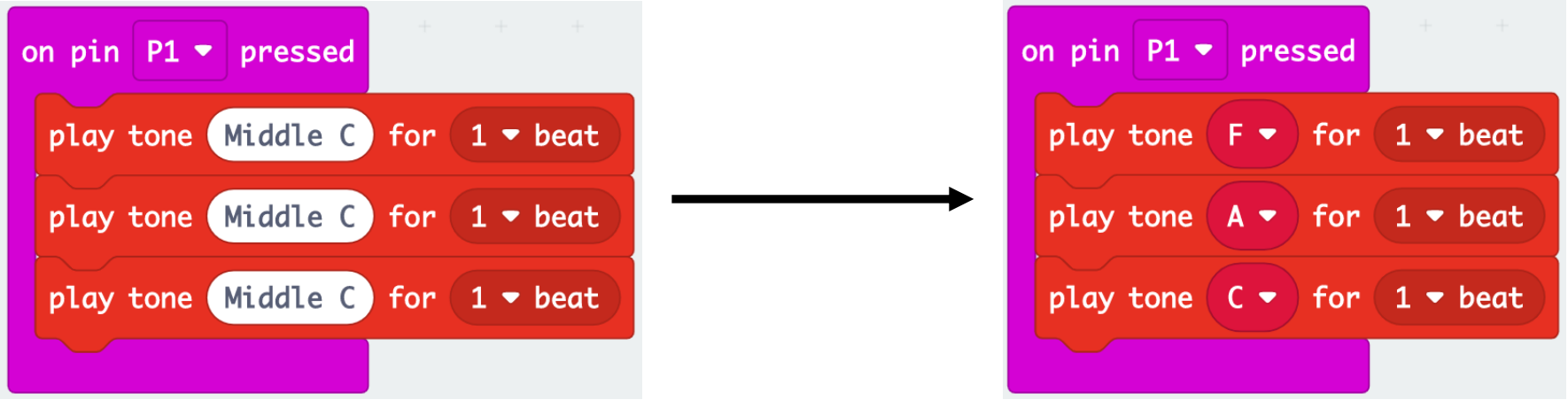
Micro:bit Octaves



1. Start by setting each of the variables to the numbers on the slide. These numbers represent the frequencies of the musical notes A, C, E, and F.

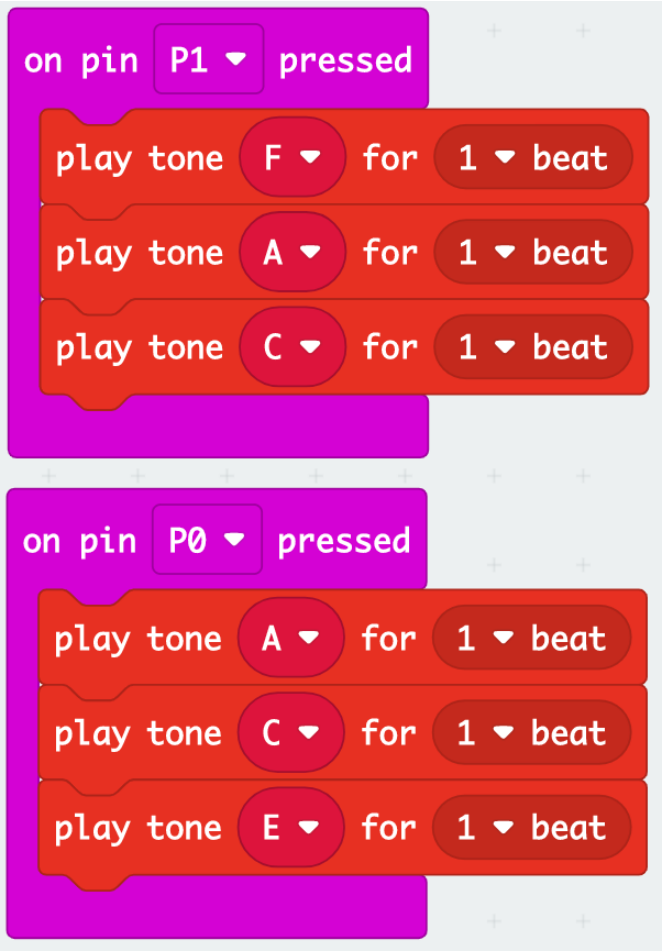
Micro:bit Octaves

2. Next, we need to tell the micro:bit what musical output to play when the pins are pressed. You have already done this when you programmed the chords. Instead of the preset notes, add your new variables to the play tone blocks.



Micro:bit Octaves

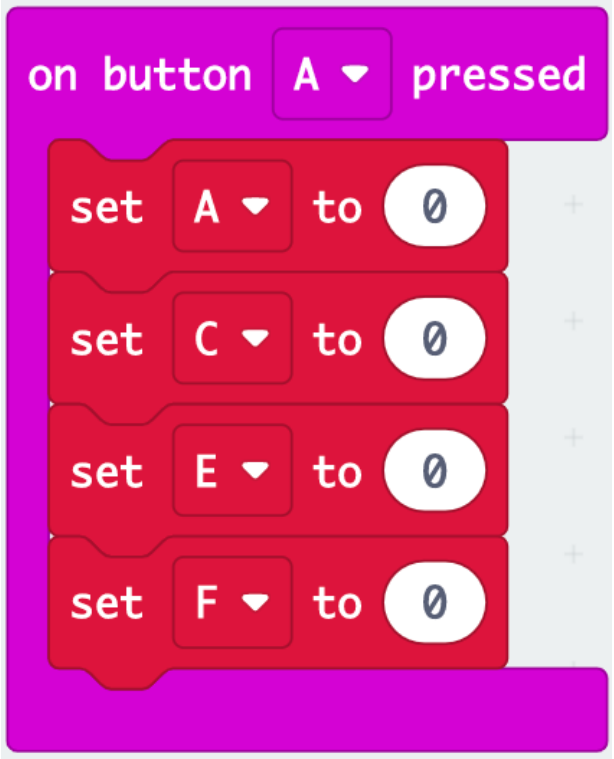
- Your code should now look like this.
- Try running the program. Does it run the same as your previous chord program?
- Next, we want to be able to change the pitch using the buttons A and B. How do you think we can do this?



Micro:bit Octaves

3. Drag the on button A pressed block from the Input section to your code.

4. Add four set variable to 0 commands to the block.

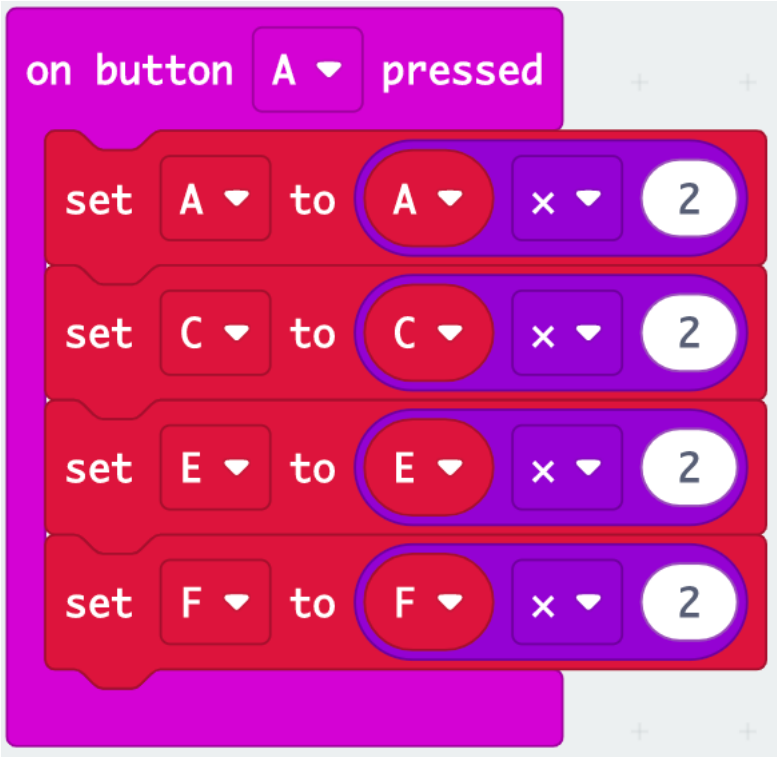


- To make the pitch higher by an octave, we need to double the frequency of the notes. How do you think we can do this?

Micro:bit Octaves

5. Drag the multiplication commands from the Math section to the set command.
6. Drag your note variables to the multiplication commands and multiply them by 2.

Can you do the the same for button B but divide the frequency by 2 instead of multiplying it?



Micro:bit Octaves

- Your program should look like this:
- Run the code and see if you can change the pitch while you play!