

For this project you will need:

- A pair of headphones or a mini speaker + instruction card
- Two crocodile clips



To use music, you first need to **import** the music library.

To do this, add this line to the top of your program: `import music`

To start with, you are going to tell the micro:bit to play one of the built-in tunes when you press button A. Check the Inputs/Outputs worksheet if you can't remember how to do this.

To play a tune, type the following code into your **if button A** statement:

```
music.play(music.ODE)
```

Connect your speaker and **test your code** – does your micro:bit play you some Beethoven?

Instead of using just button A and button B, try using the micro:bit's **accelerometer**.

It knows some built in **gestures** (see the sidebar further down the page), and uses this format:

```
accelerometer.was_gesture("up")
```

You now want to expand your program to play notes when you move the micro:bit!

To expand your **if statement**, use an **elif**, which is short for else if. For example:

```
if button_a.was_pressed():
    do something if True; otherwise, try the next statement
elif accelerometer.was_gesture("up"):
    do something if True; otherwise, try the next statement
else:
    do something else (if none of the above statements are True).
```

Gestures
"up"
"down"
"left"
"right"
"face up"
"face down"
"3g"
"6g"
"8g"
"freefall"
"shake"

You can have as many elif statements in between if and else as you want!

To play a single note, use the format:

```
music.play("A1:2")    plays the musical note A, in the lowest octave, for 2 beats
music.play("C2:4")    plays the note C, in a higher octave (try higher!), for 4 beats
```

You can use notes A-G, a # after any of these makes it sharp, and R is a rest (pause).

Expand your elif statement so that the micro:bit plays a note when held upright. Now you can use the other built-in gestures, and a selection of notes, to make a funky musical instrument!