Lesson 7 – We've done light, now action and…. Sound!

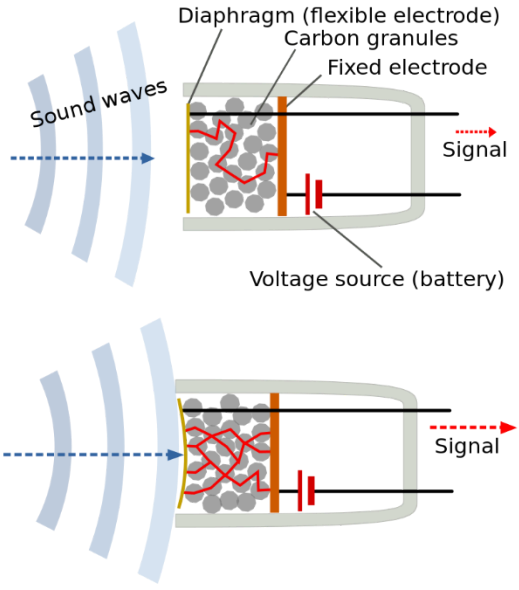
Computer and SOUND!

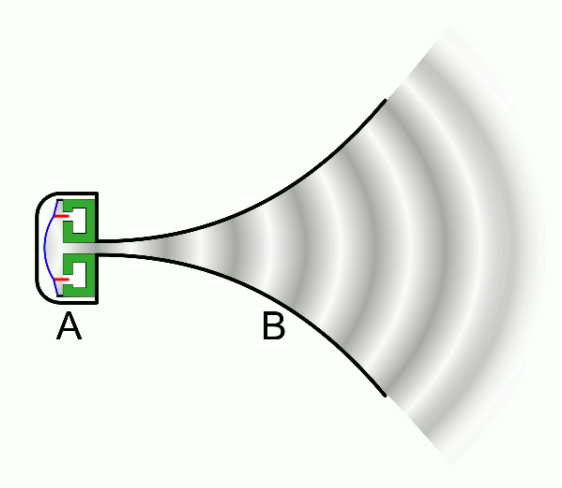
So, computers can see and interact with light, using LEDs, photoresistors and cameras… but how do they hear and talk?

Well, first we have to understand what sound really is. Sound is vibrations in the air!

Quick Quiz? Do you think you can hear things in the vacuum of space? Why or why not?

To measure sound, we MEASURE the AIR! To make sound, we need to MOVE the AIR!





The sound we hear is measured in how LOUD it is… called the Amplitude, what sound it makes, called the Frequency and how long we hear it, called the Duration.

Today we're going to learn how to make sounds using our MicroBit. The MicroBit only has one volume, so we don't have to worry about Amplitude. So, we just need to understand how to make the Frequency and the Duration.

Duration is pretty easy, that's just how long something goes. Guess how it's measured? Yep, that's right, we use seconds since that is a great way to measure time.

So what's a Frequency? Well, a frequency is how often something happens. For example, what is the Frequency of our Robotics club meeting?

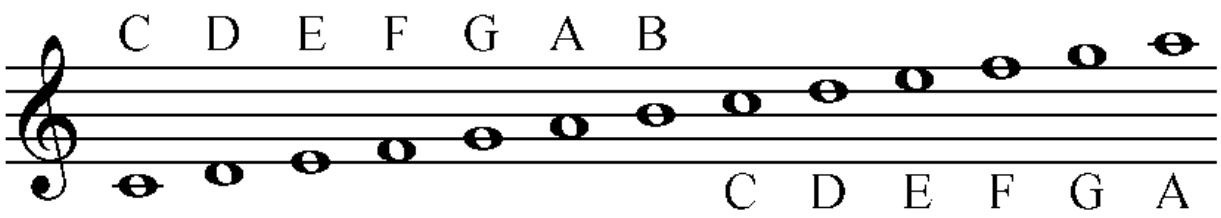
a) Once a day b) Once a week c) Once a second

If we met once a second, that would be a lot of meetings, right? Well, it turns out that's how fast we measure sound …. How many waves happen in a second! We use something called Hertz to measure frequency.

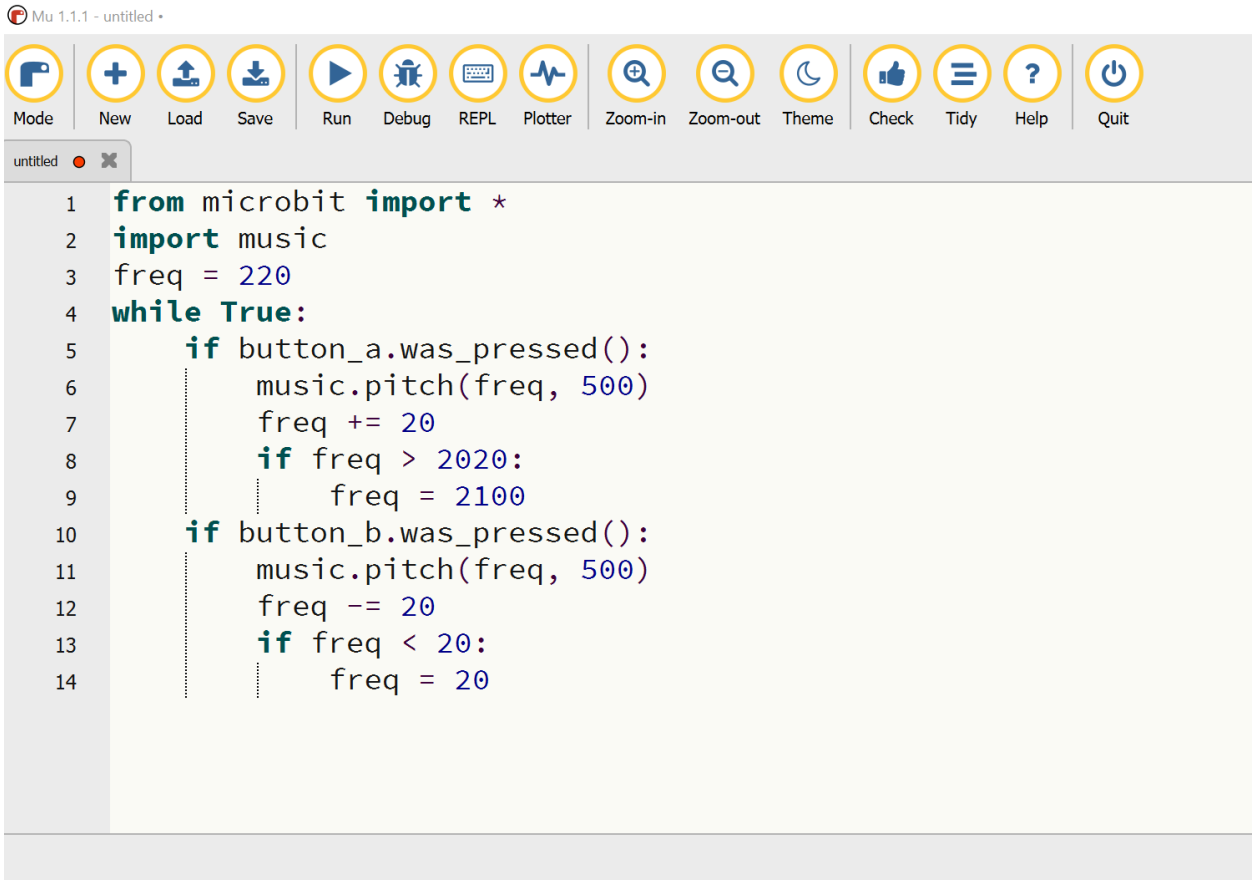
Fun Fact: If you discover something cool, like Heinrich Hertz did, you might get something named after you too! Hertz proved that electromagnetic waves exist… which means we can measure all sorts of cool things, including SOUND!

It turns out every note on the piano (or any instrument) is just something VIBRATING at a frequency! Pretty crazy, right?

	C	C#	D	Eb	E	F	F#	G	G#	A	Bb	B
5												987.8
6	1047	1109	1175	1245	1319	1397	1480	1568	1661	1760	1865	



Assignment 1: Ascending and Descending Sounds

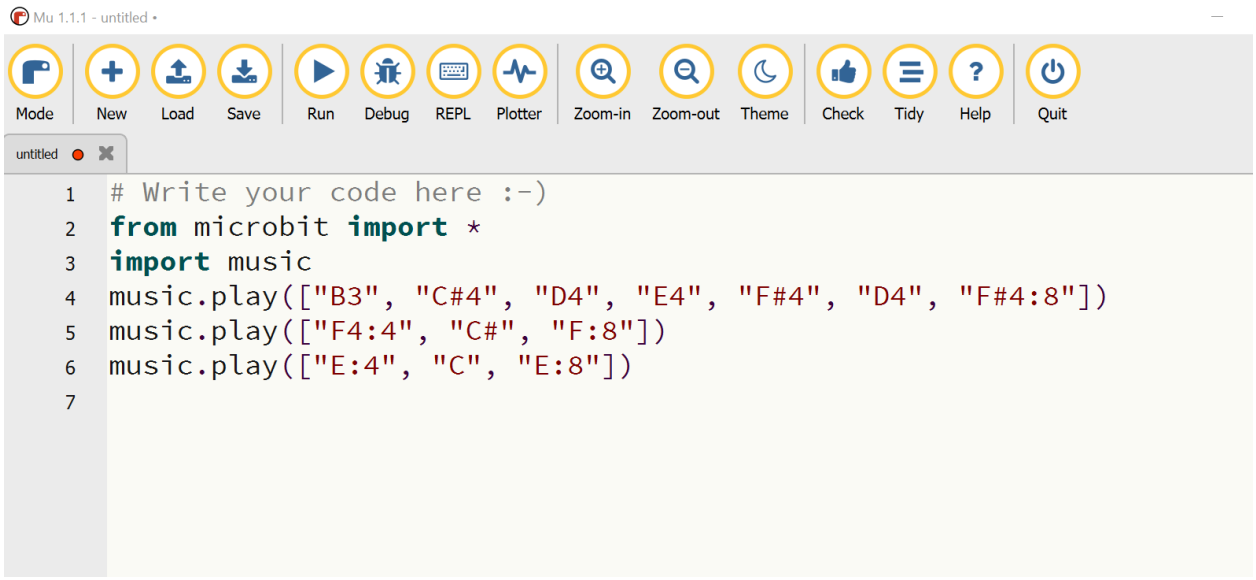



```
1  from microbit import *
2  import music
3  freq = 220
4  while True:
5      if button_a.was_pressed():
6          music.pitch(freq, 500)
7          freq += 20
8          if freq > 2020:
9              freq = 2100
10     if button_b.was_pressed():
11         music.pitch(freq, 500)
12         freq -= 20
13         if freq < 20:
14             freq = 20
```

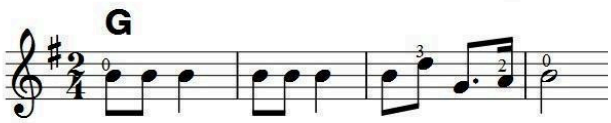
Assignment 2: Songs!

Assignment 2: Songs!

If we get more specific about our frequencies and add some durations, then we can create some sounds that are a little more fun to listen to.



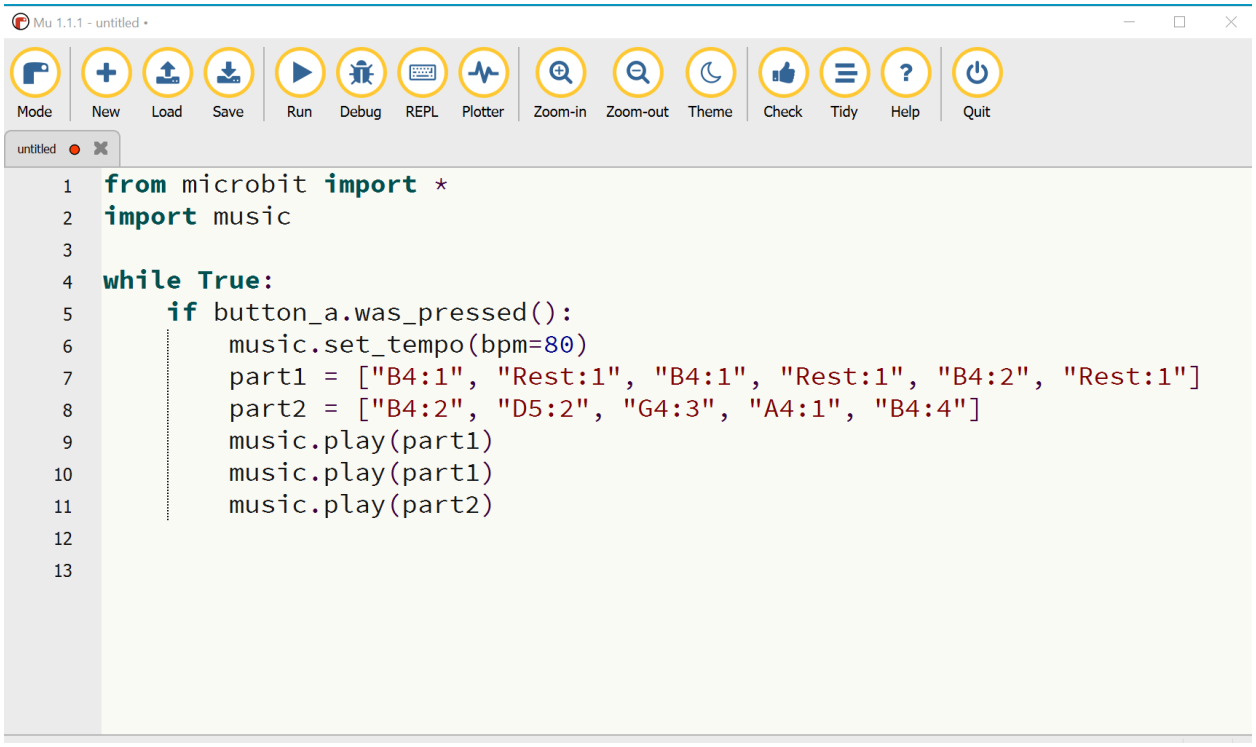
This is a song you might have heard before….



These are the notes… and their durations

B(1) B(1) B(2) B(1) B(1) B(2) B(1) D(1) G(1.5) A(0.5) B(4)

The micro:bit gets a little confused when we want to play the same frequency more than once. It smushes them all together into one long sound. To make it play the same note more than once, we have to add a silence (known in music as a REST) between the notes.)



Can you finish this song using the music below?