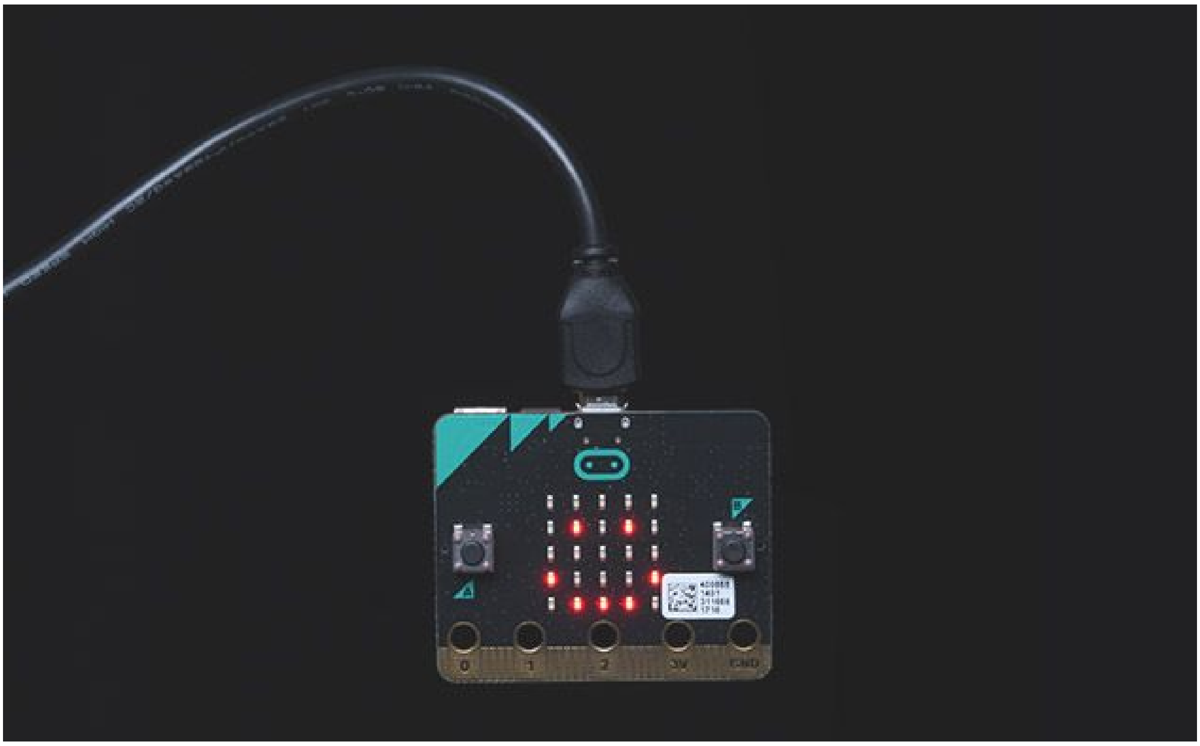
Micro:Bit Virtual Pet (Tamagotchi)



By Joe Brown 19th March 2019

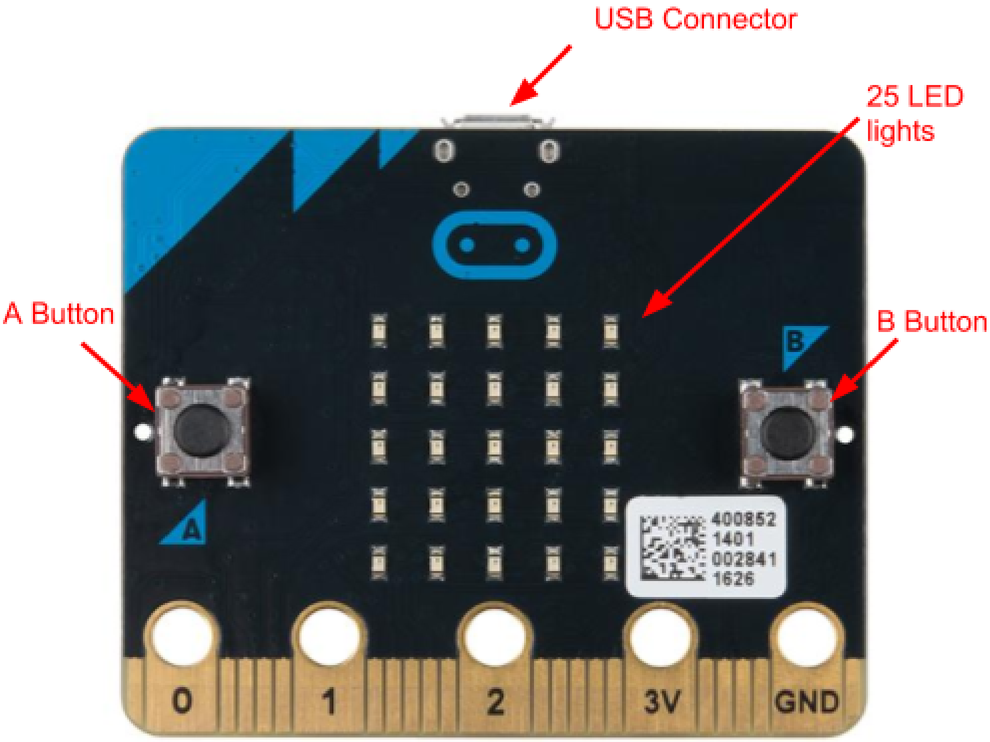
Inspired by Stu Lowe’s Workshop (Beacon Hill School)

1 Introduction

1.1 Introduction

This workshop is based on a virtual pet toy from the 1990s known as the Tamogotchi:

This is a worksheet to teach you some coding concepts using ‘block based editing’ with Micro:Bit. This is a Micro:Bit:

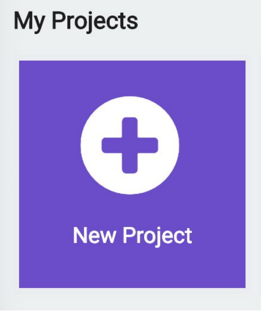


For this workshop you’ll be using the LED lights to display your pet, the buttons to interact with your pet and the USB connector to upload your code to the micro:bit.

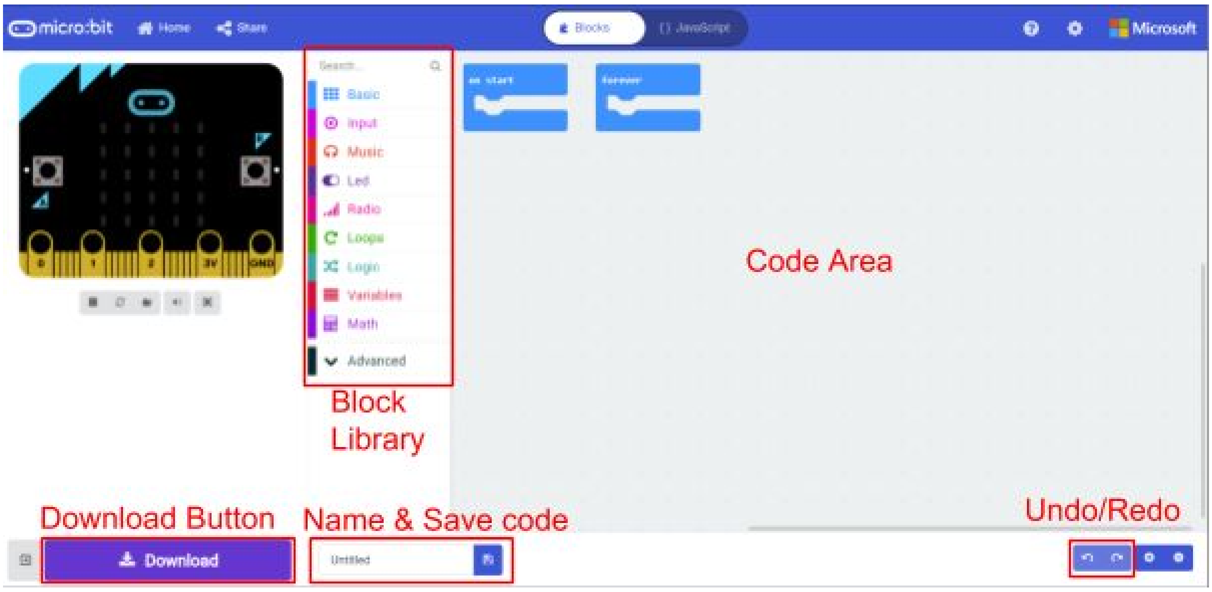
In the workshop you’re going to code the micro:bit so that over time your pet becomes unhappy when you don’t feed it, and happy when you do.

1.2 Setup

Go to https://makecode.microbit.org and click New Project:



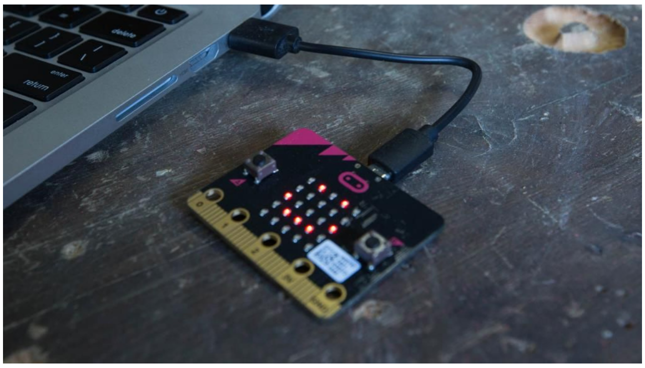
A new project will load.
This is the layout of the editor, and all the buttons you'll need:



You'll be using blocks from the block library and dragging them into the code area.

To test your code on your microbit, you'll be downloading the code with the download button.

You'll want to plug the micro:bit into your computer like so:



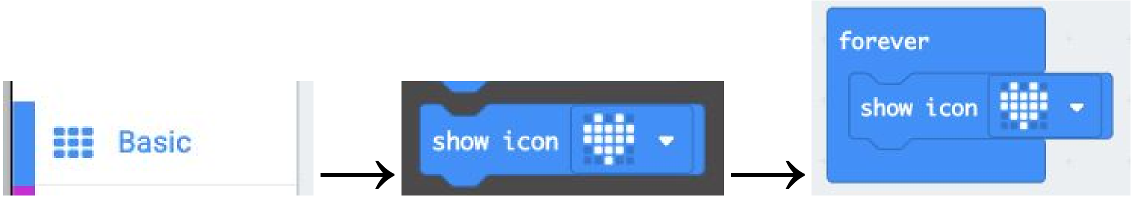
Don't worry if no lights are showing on the micro:bit yet!

2 Getting Started

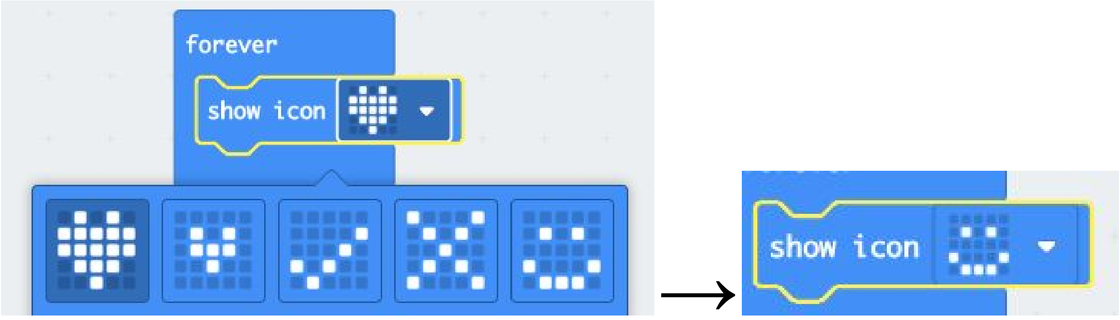
2.1 Showing your pet's face

We'll be showing your pet on the micro:bit 'screen' which is a grid of 25 LED lights which will represent your pet's face.

Let's test that out now. Click on the **Basic** tab, click and drag the **show icon** block into the **forever** block:



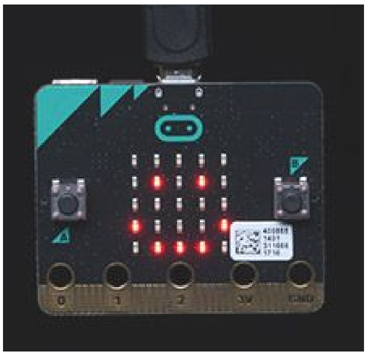
Now change the icon by using the drop down to a smile emoji like so:



2.2 Testing the code

Great now let's test our code! Plug the USB cable into the computer and into the micro:bit Click the download button and the bottom of the screen and save the code (hex file) to the Micro:Bit.

You may need to find it on 'My Computer' called MICROBIT - Ask for help if you're stuck. You may need to copy your hex file from Downloads to MICROBIT.
After the hex file has copied to the Micro:Bit, you should see a smiley face on the lights:



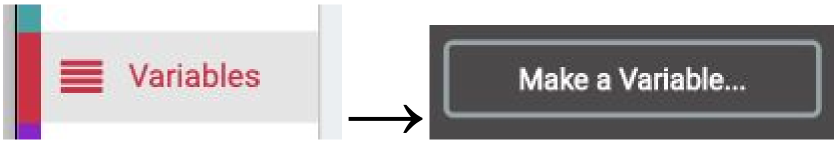
If not ask for help.

3 Pet Health & Happiness

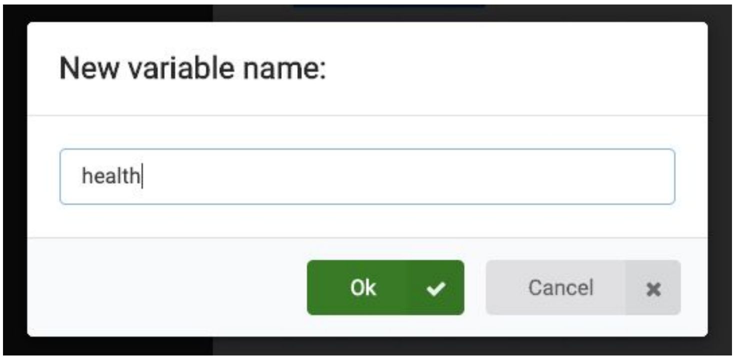
3.1 Pet Health Variable

Now we're going to create a 'variable' called health.
A variable is a thing that is remembered by the micro:bit - in this case it's going to be a number for health. Where 10 is most happy and 0 is most sad (the pet needs feeding!)

So let's make the variable for health. Click the **Variables** tab and **Make a Variable…** button:



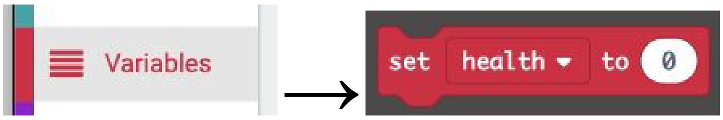
Name it 'health' and click ok:



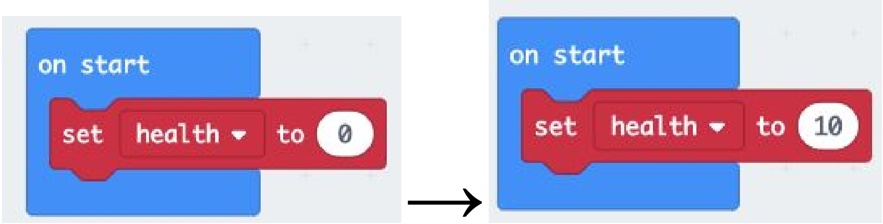
New variable name:

health

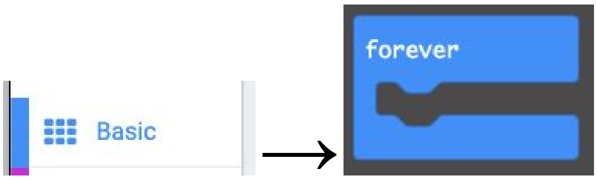
So let's start with a health of 10 when the code starts on the micro:bit.
Click the **Variables** tab and then **set health to 0** block:



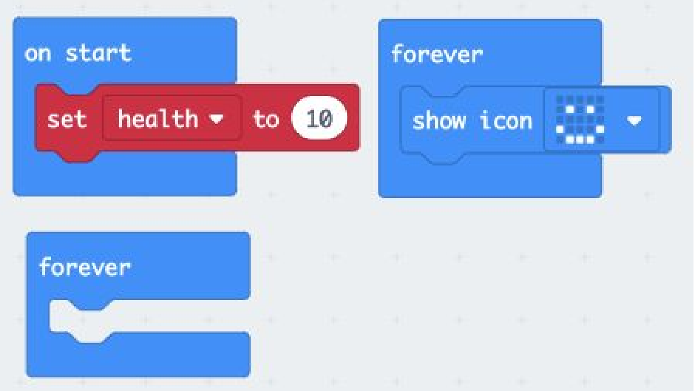
Click and drag the set **set health to 0** block into the **on start** block and change the 0 to 10:



So when the code starts on the micro:bit, the pet's health is 10. We want to make this slowly go down over time. So, let's create a new **forever** block from the **basic** tab:

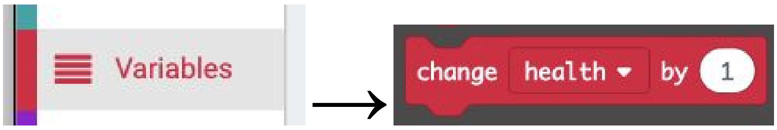


So your code looks like this:



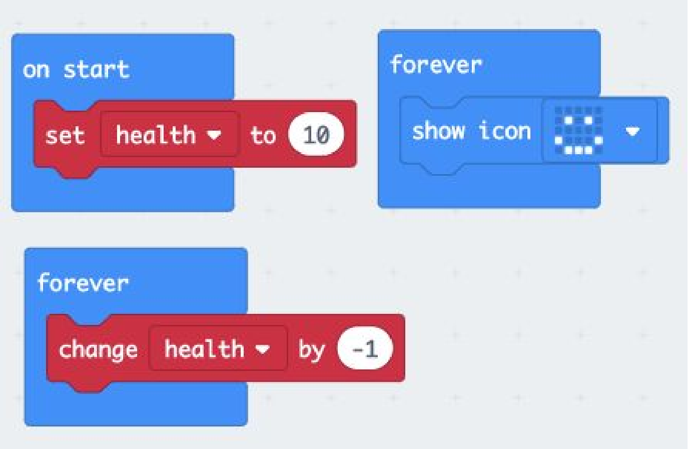
```
on start
    set health to 10

forever
    show icon

forever
```

To lower health, we need a **change health by 1** block from the **Variables** tab:



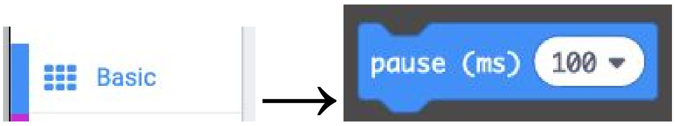
```
change health by 1
```

Put this in your new **forever** block and change the value to -1:



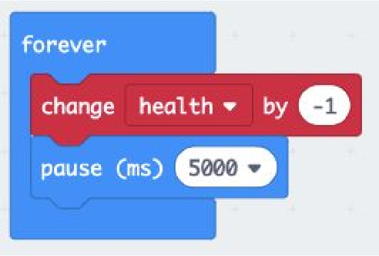
```
on start
    set health to 10

forever
    show icon

forever
    change health by -1
```

However the forever loop happens really quickly as computers think fast! So we need to make the health drop slowly. We can do this with a **pause** block:



```
pause (ms) 100
```

‘ms’ means milliseconds - there are 1000 milliseconds in a second.
We want health to lower by one every 5 seconds, so we want to set the pause to 5000:



```
forever
    change health by -1
    pause (ms) 5000
```

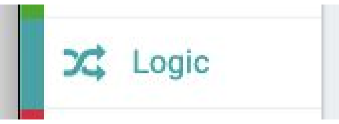
Next up we need to display the health on the micro:bit lights to show how happy the pet is.

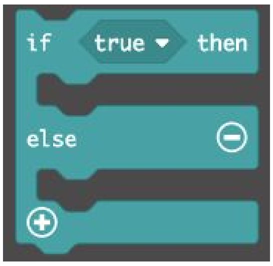
3.2 Pet happiness and If Statements

So currently every 5 seconds the health of our pet lowers by 1, starting from 10. We want to show how happy our pet is like so:

- **If** health is more than 6 then happy
- **Else if** health is less than or equal to 6 **and** health is more than 3 then neutral
- **Else if** health is less than or equal to 3 **and** health is more than 1 then unhappy
- **Else** sad

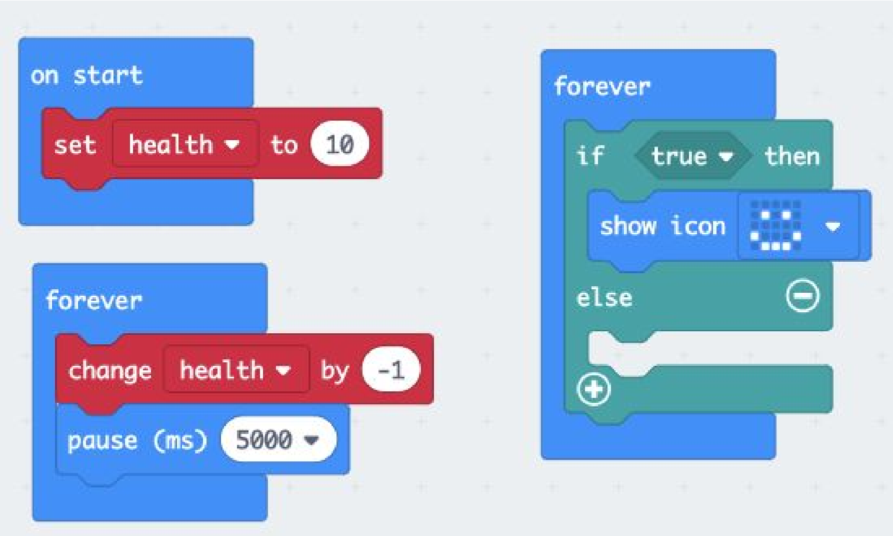
Micro:bit has a block for **if, else if, else** known as "if statements" - If this then do this! Get that block from the **logic** tab:





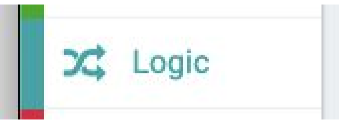
```
if true then

else
```

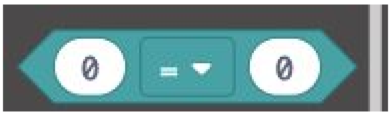
Drag it to our **forever** block that has the smiley face:



```
on start
    set health to 10

forever
    change health by -1
    pause (ms) 5000

forever
    if true then
        show icon
    else
```

Now, we want to make this code: If health is more than 6 then happy
We need the comparison block from the **logic** tab:

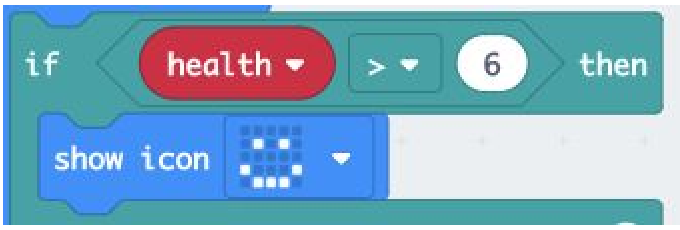




```
0 = 0
```

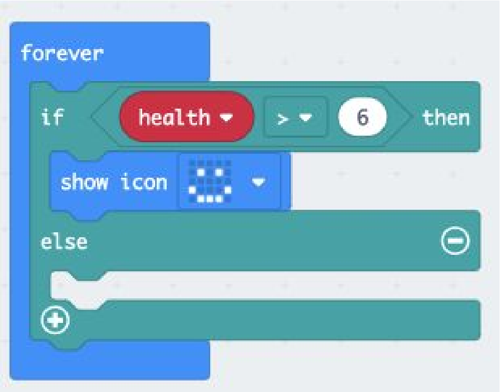
Drag this into the if block.

Now change the = to a >, add the health variable and change the 0 to a 6!



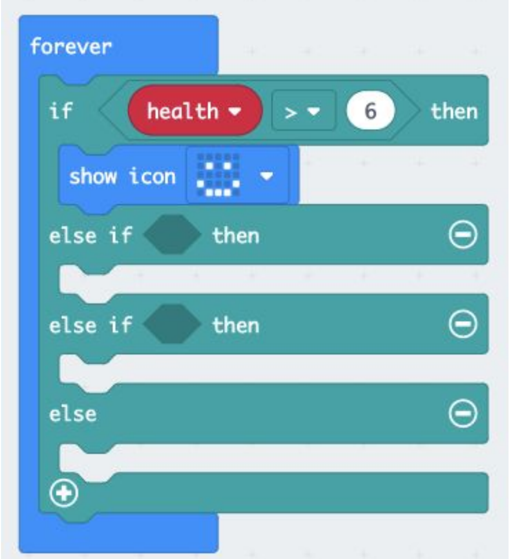
```
if health > 6 then
    show icon
```

Now we need to add the rest of the if statements.
Click the plus button on the **if** block to add else ifs and fill out the code!



```
forever
  if health > 6 then
    show icon
  else
```

→

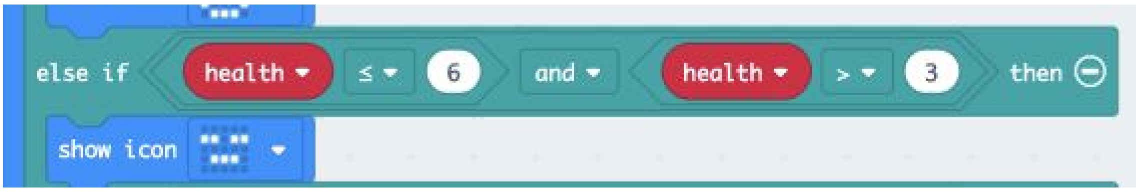


```
forever
  if health > 6 then
    show icon
  else if then
  else if then
  else
```

- **Else if** health is less than or equal to 6 **and** health is more than 3 then neutral

<u>TIPS:</u>
≤ is the symbol for less than or equal to.
And blocks are in the **logic** tab.
You can copy blocks or groups of blocks with right click duplicate.

Here's the first **Else if**:



```
else if health ≤ 6 and health > 3 then
  show icon
```

See if you can code the rest of the statements:

- **Else if** health is less than or equal to 3 **and** health is more than 1 then unhappy
- **Else** sad

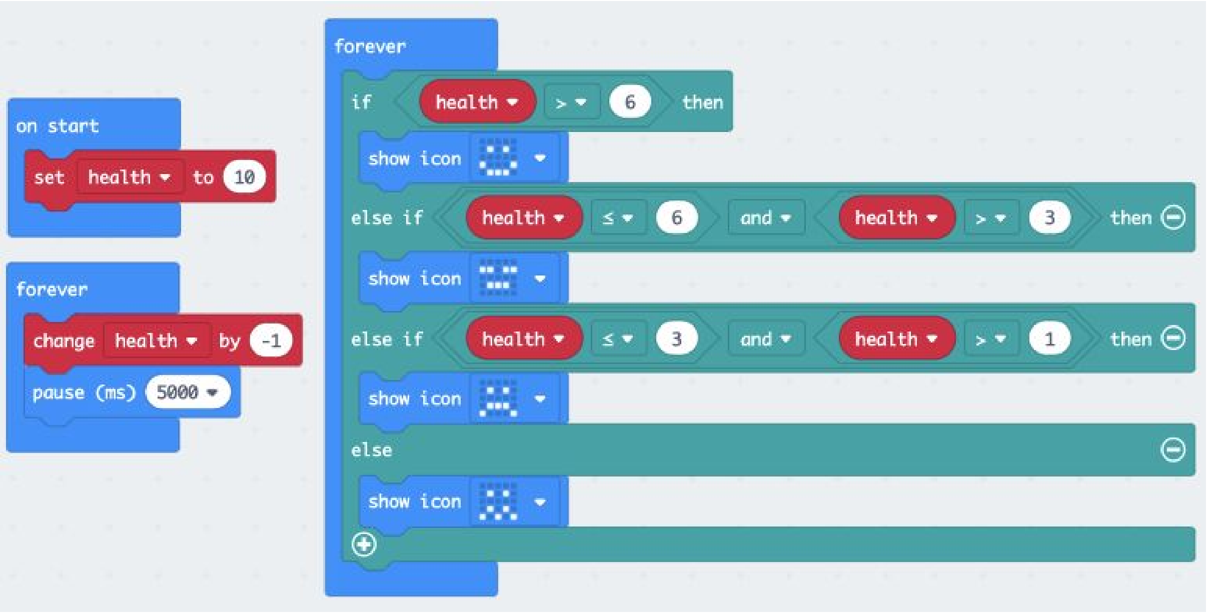
There's the full solution on the next page, but try to code these two statements without looking first!

Once you've filled this in, let's test it.
Download the code to the micro:bit and test it!

Hopefully you should see slowly over time your pet gets sadder. Remember it will take 20 seconds before the first change.
If you're stuck ask for help, and if you need to look the solution is on the next page.

To reset it, or test again, there's a reset button on the back of the microbit!
If you want to see the changes go faster, make the pause shorter (maybe 1000ms)

Solution for happiness levels:

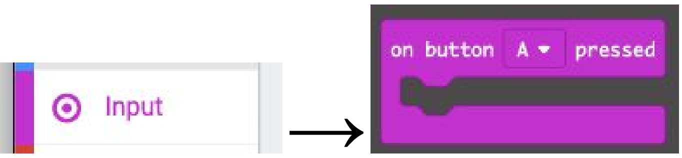


```
on start
  set health to 10

forever
  change health by -1
  pause (ms) 5000

forever
  if health > 6 then
    show icon
  else if health ≤ 6 and health > 3 then
    show icon
  else if health ≤ 3 and health > 1 then
    show icon
  else
    show icon
```

3.3 Feeding your pet

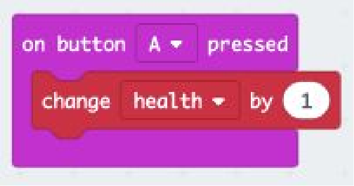
It’s sad to see your pet get unhappy! We can help this by coding a way to feed your pet.

In the **Input** tab there’s a block **called on button A pressed:**



```
on button A pressed
```

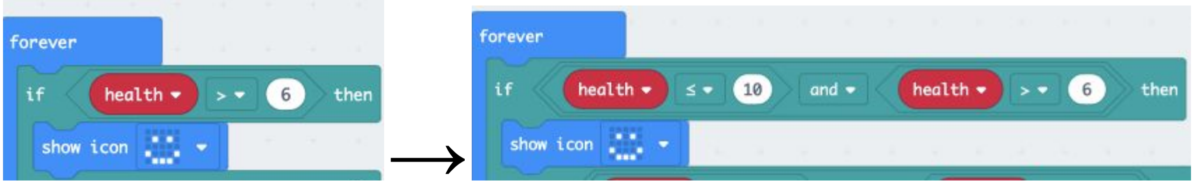
Let’s make it so when you press the A button on the micro:bit, it increases happiness by 1:



```
on button A pressed
  change health by 1
```

Now download and test your code - If your micro:bit pet gets unhappy you should be able to make it happier by pressing the A button!

But pets can’t eat infinite food! Give them too much food and they get unhappy, we can fix that by changing the first if statement:



```
forever
  if health > 6 then
    show icon
```

```
forever
  if health ≤ 10 and health > 6 then
    show icon
```

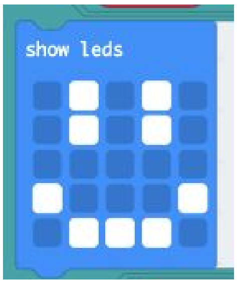
Download and test the code - If you overfeed your pet (go above health of 10) the **else** sad statement will be true. So if health = 11, the sad face will show.
Don’t worry, over time it will be happy again once health drops back down to 10.

4 Extensions

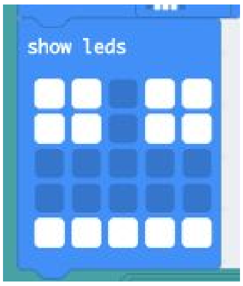
Now you’ve finished the main workshop, here’s some extras you can add:

4.1 Customise Faces

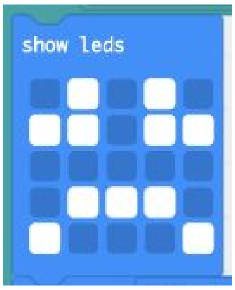
Instead of using **show icon** blocks for faces, try **show leds** (in the **basic** tab) you can create your own custom faces! Here’s some examples:



```
show leds
```



```
show leds
```

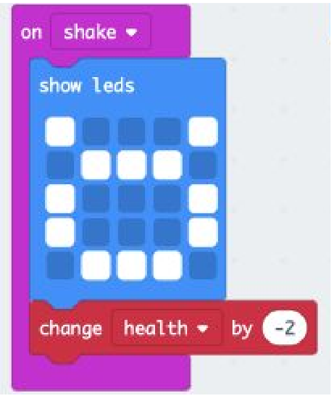


```
show leds
```

4.2 Other Inputs

You can make other inputs

In the **input** tab there’s a **on shake** block, which you could make the health go down quickly if you shake the micro:bit for example:



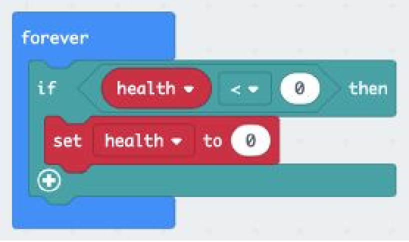
```
on shake
  show leds
  change health by -2
```

This will briefly display a ‘:O’ emoji as you shake and reduce health by 2!

- There’s other inputs than shake such as tilting, dropping, face down… have a look!
- There’s also the B button you could use for ‘treats’ that increase health quicker than the A button for example.

4.3 Health getting too low

You may have noticed currently after a while health drops so low that it takes so many button presses to get your pet happy again! (say if happiness -20, so you’d have to press A 21 times to change the face) You can fix that by limiting lowest health to 0:



```
forever
  if health < 0 then
    set health to 0
```