Make your own LED light show

Create a basic LED light show with different images and animations using the BBC micro:bit. Pressing button A allows you to cycle through the images and animations you've programmed, displaying it on the BBC micro:bit's LED matrix.

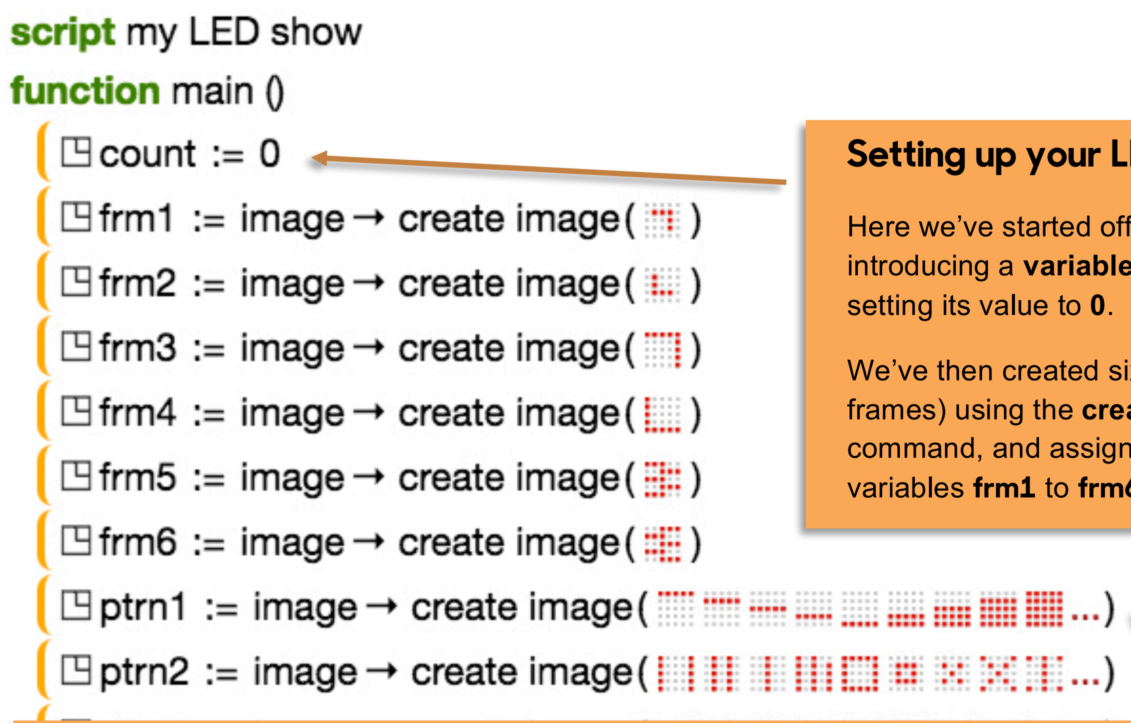
Step 1: Import the code

Click on the hex file link on the Live Lessons website to view the code on the BBC micro:bit website (www.microbit.co.uk).

The script for your animation should now appear in your code window.

Hit 'run' to see it in action on the simulator, or plug in your BBC micro:bit, hit 'compile' and drag the hex file onto your micro:bit. Press button A and see the LED light show play.

Step 2: Understanding the code



```
script my LED show
function main ()
    count := 0
    frm1 := image → create image( )
    frm2 := image → create image( )
    frm3 := image → create image( )
    frm4 := image → create image( )
    frm5 := image → create image( )
    frm6 := image → create image( )
    ptrn1 := image → create image( ...)
    ptrn2 := image → create image( ...)
```

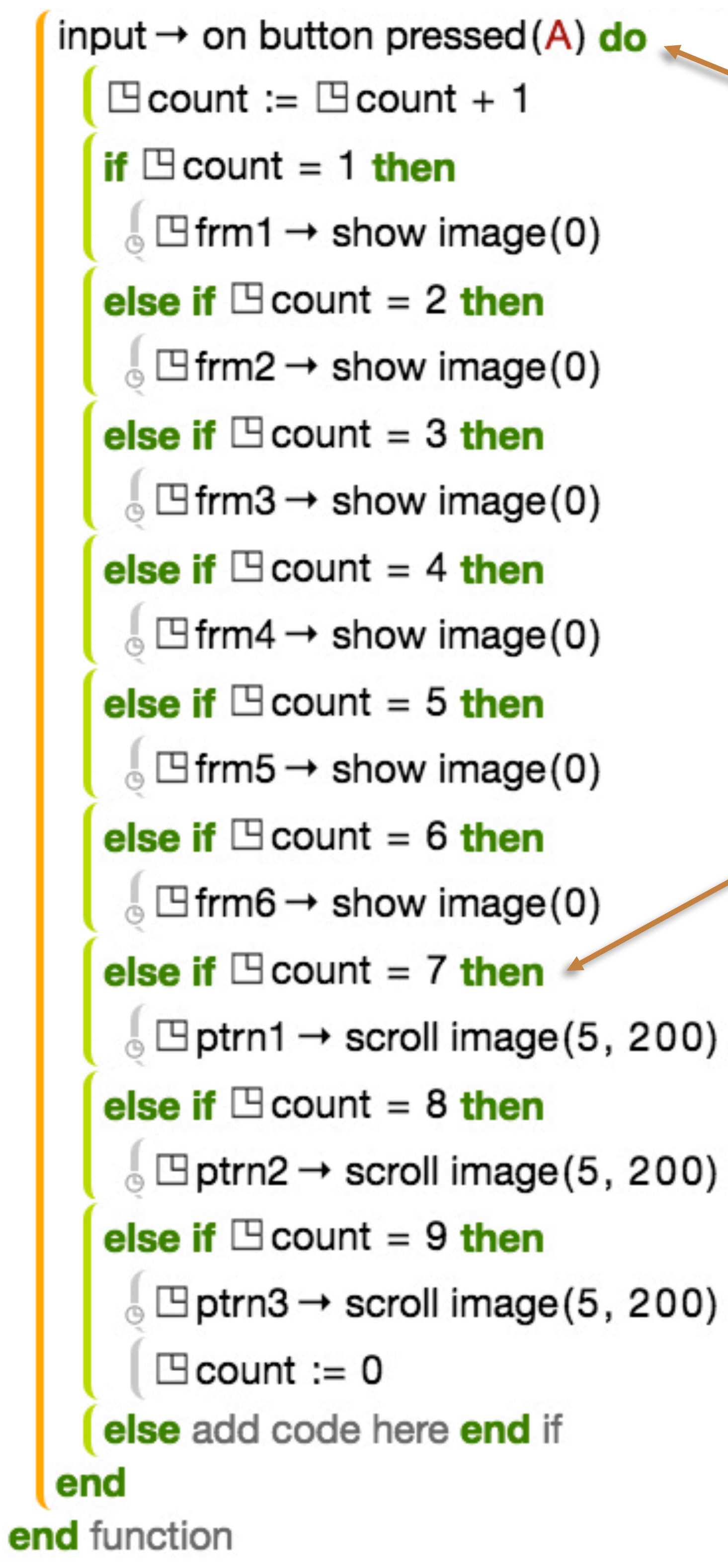
Setting up your LED light show

Here we've started off the program by introducing a **variable** called **count**, and setting its value to **0**.

We've then created six images (or frames) using the **create image** command, and assigned them to six variables **frm1** to **frm6**.

Designing your animations

We've also created three animations (or patterns) using the same **create image** command and adding additional frames to the image. We've assigned these three animations to the variables **ptrn1** to **ptrn3** respectively.


```
input → on button pressed(A) do
    count := count + 1
    if count = 1 then
        frm1 → show image(0)
    else if count = 2 then
        frm2 → show image(0)
    else if count = 3 then
        frm3 → show image(0)
    else if count = 4 then
        frm4 → show image(0)
    else if count = 5 then
        frm5 → show image(0)
    else if count = 6 then
        frm6 → show image(0)
    else if count = 7 then
        ptrn1 → scroll image(5, 200)
    else if count = 8 then
        ptrn2 → scroll image(5, 200)
    else if count = 9 then
        ptrn3 → scroll image(5, 200)
        count := 0
    else add code here end if
end
end function
```

What happens when you press button A?

Here we've said then when button A is pressed, we add **1** to the value of the variable **count**.

That means that when you first press A, the value of **count** is **1**, and the second time you press A, the value of **count** is **2**, and so on.

We've then introduced some conditional statements. IF **count** is **1**, then the BBC micro:bit displays the image stored in **frm1**, IF **count** is **2**, then the BBC micro:bit displays the image stored in **frm2**, and so on.

Displaying images as animations

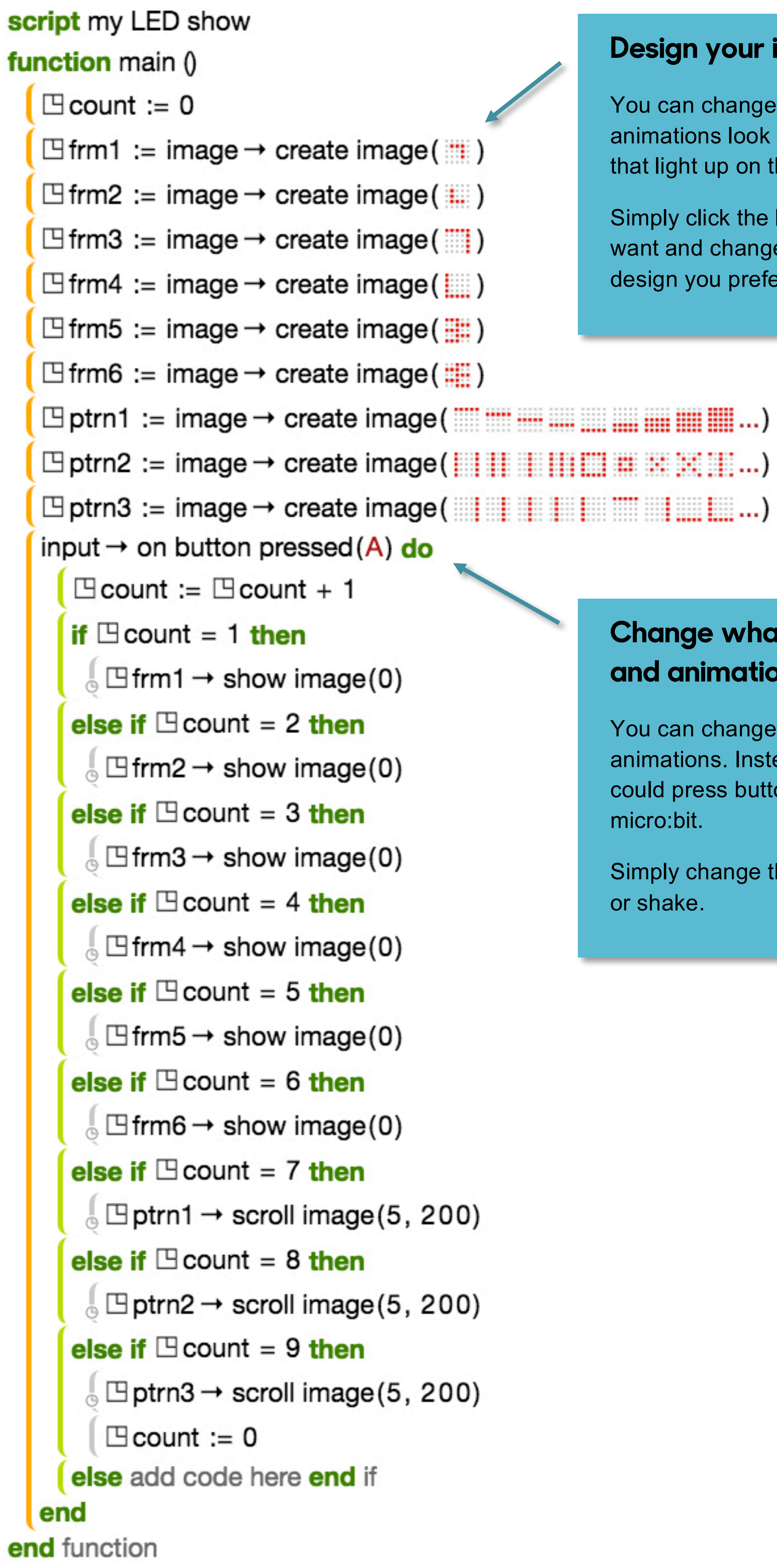
When we come to displaying the animations (or patterns), instead of the **show image** command, we use the **scroll image** command.

Instead of the images scrolling across in a continuous stream, however, we want it to display frame by frame, so we've added an offset of the x-coordinates of 5. We've also said that the interval between each frame should be 200 milliseconds.

When count gets to 9, we want to start again from the beginning with the first frame, so we state that count is back to 0.

Step 3: Changing the code

Turn over the page to get some ideas of how you can change the code to make the LED light show your own.



```
script my LED show
function main ()
    count := 0
    frm1 := image → create image( )
    frm2 := image → create image( )
    frm3 := image → create image( )
    frm4 := image → create image( )
    frm5 := image → create image( )
    frm6 := image → create image( )
    ptrn1 := image → create image( ...)
    ptrn2 := image → create image( ...)
    ptrn3 := image → create image( ...)
    input → on button pressed(A) do
        count := count + 1
        if count = 1 then
            frm1 → show image(0)
        else if count = 2 then
            frm2 → show image(0)
        else if count = 3 then
            frm3 → show image(0)
        else if count = 4 then
            frm4 → show image(0)
        else if count = 5 then
            frm5 → show image(0)
        else if count = 6 then
            frm6 → show image(0)
        else if count = 7 then
            ptrn1 → scroll image(5, 200)
        else if count = 8 then
            ptrn2 → scroll image(5, 200)
        else if count = 9 then
            ptrn3 → scroll image(5, 200)
            count := 0
        else add code here end if
    end
end function
```

Design your images and animations

You can change what your images and animations look like by changing the pixels that light up on the LED matrix.

Simply click the line of code for the image you want and change the pixels to whatever design you prefer.

Change what triggers the image and animations

You can change what triggers the image and animations. Instead of pressing button A, you could press button B, or shake your BBC micro:bit.

Simply change the input to pressing button B or shake.

Test, play and show us what you’ve done

Now that you’ve made your very own LED light show, click ‘run’ to test it on the simulator and ‘compile’ to see it working on your micro:bit.

Click ‘export’ to save off your code and send it to us at **live.lessons@bbc.co.uk**. You could see your animations featured on our **Strictly micro:bit Live Lesson** on the 24th of March.