LED strips – zoom from left to right

This guide builds on previous tutorials.

You will learn how to use While and For loops to make LEDS flash in sequence.

Making your strip light up right to left

This code will make your LED strip light up one at a time:

Count the number of LEDs on your LED strip. There are 14 in this example, but you may have 10.

We have imported the utime library here. It is designed for micropython. But it is very similar to the Time library so you could write **import time**.

```
1  from neopixel import Neopixel
2  import utime
3
4  numpix = 14
5  strip = Neopixel(numpix, 0, 28, "RGB")
6
7  red = (255, 0, 0)
8  delay = 0.5
9
10 strip.brightness(42)
11 blank = (0,0,0)
12
13 while True:
14
15     for x in range(13):
16         strip.set_pixel(x+1, red)
17         strip.show()
18         utime.sleep(delay)
```

Copy the code below into Thonny and run it:

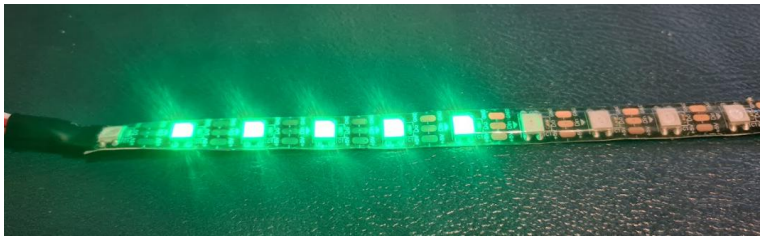
```
from neopixel import Neopixel
import utime
numpix = 14
strip = Neopixel(numpix, 0, 28, "RGB")
red = (255, 0, 0)
delay = 0.5
strip.brightness(42)
blank = (0,0,0)
while True:
    for x in range(13):
        strip.set_pixel(x+1, red)
        strip.show()
        utime.sleep(delay)
```

Did your LED strip light up?

Did each pixel light up, one at a time?

What colour were the LEDs?

This was the outcome for me:



I took the picture as the LEDs were lighting up.

But why are they outputting a green light when we have used RGB values for red?

Why are the LEDs green?

In the 1.4 guide we noted that the RGB values for 'red', whilst correct, were outputting a green light.
The same is happening here.

```
5  red = (255, 0, 0)
```

These RGB values *are* correct for the colour red.
For some reason some LED strips are wired a little differently.
Make it so the 'red' variable outputs a red colour.
You can fix this by changing this code:

```
strip = Neopixel(numpix, 0, 28, "GRB")
```

Instead of "RGB" I have swopped it, so it now shows "GRB":

Challenge:

Try creating a new colour variable under this line of code:

```
7  red = (255, 0, 0)
```

You can find out the RGB valies for colours using this web page: [HTML Color Picker (w3schools.com)]

This is the RGB value for purple:

```
rgb(255, 0, 255)
```

Making your strip light up left to right, one at a time

Adding this line of code makes each pixel light up, then go blank before the loop is repeated.

```
13  while True:
14
15      for x in range(13):
16          strip.set_pixel(x+1, red)
17          strip.show()
18          utime.sleep(delay)
19          strip.set_pixel(x+1, blank)
```

Making your strip light up zooming from left to right, then right to left

Adding this code at the end makes a new for loop to make the pixels light up, going the other way:

```
21      for x in reversed(range(13)):
22          strip.set_pixel(x+1, red)
23          strip.show()
24          utime.sleep(delay)
25          strip.set_pixel(x+1, blank)
26
```

You can copy this code into Thonny:

```
for y in reversed(range(13)):
    strip.set_pixel(y+1, red)
    strip.show()
    utime.sleep(delay)
    strip.set_pixel(y+1, blank)
```

This is how my Thonny program looks:

The utime library is almost exactly the same as the time library (but is designed for microcontrollers like the pico). You could use either library here.

If your LEDs are green instead of red, try changing these RBG values.

Change this value to go faster.

This 'blank' variable sets the RGB values for making NO light emit from the LED.

This is 13 because 0 to 13 = 14 LEDs.

```
1   from neopixel import Neopixel
2   import utime
3   numpix = 14
4   strip = Neopixel(numpix, 0, 28, "RGB")
5   red = (255, 0, 0)
6   delay = 0.5
7   strip.brightness(42)
8   blank = (0,0,0)
9
10  while True:
11      for x in range(13):
12          strip.set_pixel(x+1, red)
13          strip.show()
14          utime.sleep(delay)
15          strip.set_pixel(x+1, blank)
16
17      for y in reversed(range(13)):
18          strip.set_pixel(y+1, red)
19          strip.show()
20          utime.sleep(delay)
21          strip.set_pixel(y+1, blank)
22
```

Challenge:

- ✓ Change the colours
- ✓ Change the rate at which the LEDs flash
- ✓ Make 3 pixels light up on each flash
- ✓ Make the two outside pixels (of the 3 that are lit) be less bright