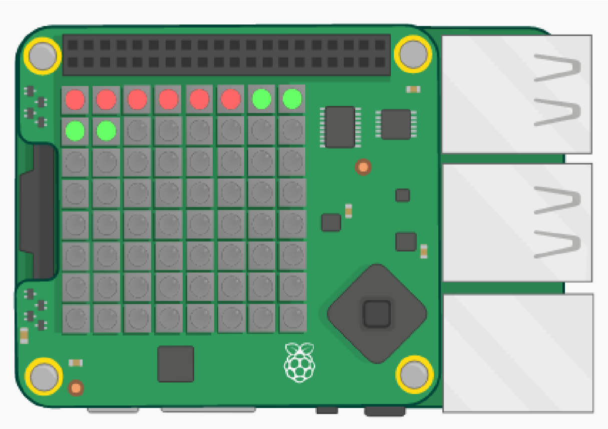
Countdown timer

Introduction

In this project you will use the coloured pixels on the Sense HAT to display a countdown timer.

What you will make

Click **Run** to start the countdown timer:



What you will learn

This project covers elements from the following strands of the Raspberry Pi Digital Making Curriculum (http://rpf.io/curriculum):

- Apply basic programming constructs to solve a problem (https://www.raspberrypi.org/curriculum/programming/builder)

Additional information for educators

If you need to print this project, please use the printer-friendly version (https://projects.raspberrypi.org/en/projects/countdown-timer/print).

Use the link in the footer to access the GitHub repository for this project, which contains all resources (including an example finished project) in the 'en/resources' folder.

What you will need

Hardware

- A computer capable of accessing the trinket.io (https://trinket.io) website

Software

This project can be completed in a web browser using trinket.io (https://trinket.io).

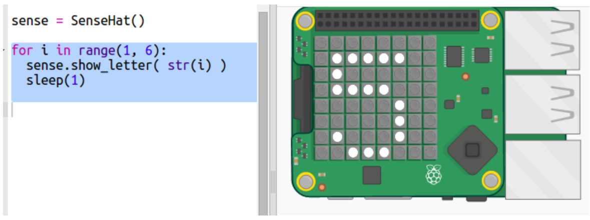
Text countdown

First let's count down from 5 to 0 by displaying numbers using the Sense HAT's pixel display.

- Open the Countdown timer starter trinket: jumpto.cc/timer-go (http://jumpto.cc/timer-go)

  The code to set up the Sense HAT has been included for you.

- You're going to count up to 5 first because that's easier to do. Add the highlighted code to the bottom of your script:



```
sense = SenseHat()

for i in range(1, 6):
  sense.show_letter( str(i) )
  sleep(1)
```

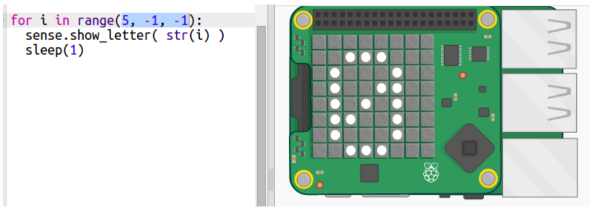
The command `sense.show_letter()` displays a single letter on the Sense HAT. It doesn't allow numbers, so you have to use `str()` to change the number into a format that it can display.

`sleep(1)` waits one second before the code moves on to the next step.

- In Python, `range(1, 6)` returns the numbers 1 through to 5. You don't have to count in ones though:

  - range(1, 10, 2) would count up in twos, giving 1, 3, 5, 7, and 9
  - range(5, 0, -1) counts down by taking away -1, giving 5, 4, 3, 2, 1

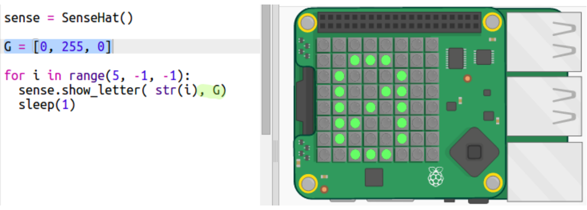
  Change the range in your code so that it counts down to 0:



```
for i in range(5, -1, -1):
  sense.show_letter( str(i) )
  sleep(1)
```

- The number on the LEDs doesn't have to be white — the Sense HAT can display lots of colours. It uses RGB colours (red, green, and blue).

  Try using green:

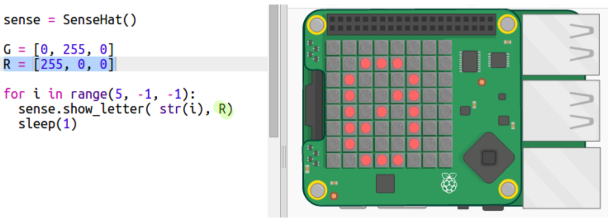


```
sense = SenseHat()

G = [0, 255, 0]

for i in range(5, -1, -1):
  sense.show_letter( str(i), G)
  sleep(1)
```

Challenge: another colour

Can you change the colour to one you like?

Here's another example that uses the colour red:



```
sense = SenseHat()

G = [0, 255, 0]
R = [255, 0, 0]

for i in range(5, -1, -1):
  sense.show_letter( str(i), R)
  sleep(1)
```

Try experimenting with the R, G, and B values — you can use any number from 0 to 255). What colour is `[255, 0, 255]`?

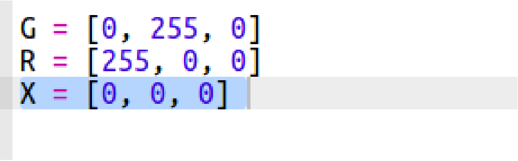
You can also look up the RGB values for a specific colour here (http://jumpto.cc/colours).

Creating a dot timer

Another way to create a timer is by turning pixels from green to red.

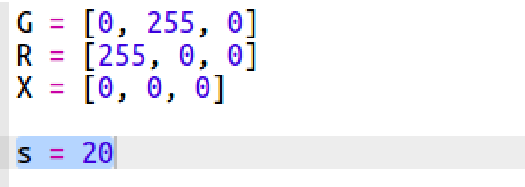
- Open the dot timer starter trinket: jumpto.cc/dot-timer-go (http://jumpto.cc/dot-timer-go)

- Add a variable `X` to use to turn pixels off — it has no red, green, or blue in its RGB value:



```
G = [0, 255, 0]
R = [255, 0, 0]
X = [0, 0, 0]
```

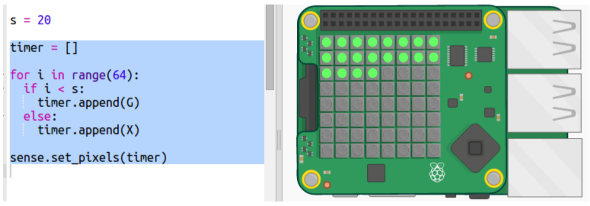
- Add a variable called `s` for the number of seconds you want to count.



```
G = [0, 255, 0]
R = [255, 0, 0]
X = [0, 0, 0]

s = 20
```

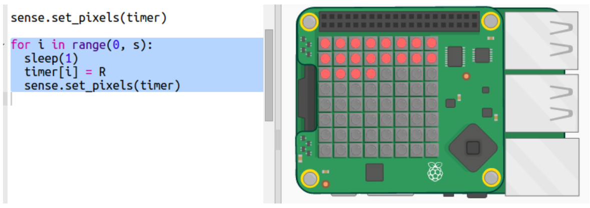
- You can give the Sense HAT a list of 64 (8×8) colours to display, starting from the top left and working down a row at a time.

  Let’s create a list of colours by creating a green pixel dot for each second we want to count, and setting the rest of the 64 pixels so they are off. The `timer` variable contains the list of colours to display and starts off empty:



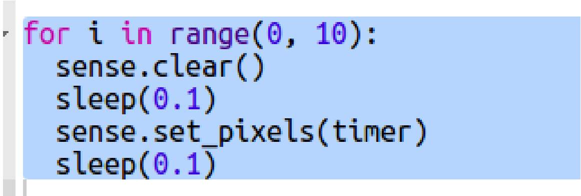
```
s = 20

timer = []

for i in range(64):
  if i < s:
    timer.append(G)
  else:
    timer.append(X)

sense.set_pixels(timer)
```

- Now let’s run the countdown by turning a pixel red every second:



```
sense.set_pixels(timer)

for i in range(0, s):
  sleep(1)
  timer[i] = R
  sense.set_pixels(timer)
```

- You could also flash the display **at the end** by turning the pixels on and off:



```
for i in range(0, 10):
  sense.clear()
  sleep(0.1)
  sense.set_pixels(timer)
  sleep(0.1)
```

Challenge: timer games

- Can you create a timer for a game or challenge? Think about whether you timers needs to count up or down, and about what colours it should have.

Use your timer to challenge a friend — one of you should watch the timer while the other completes the challenge.

Use one of these ideas as a challenge, or come up with your own:

- Can you recite the alphabet in 5 seconds? What about backwards?
- How many times can you type ‘Code Club’ in 10 seconds?
- Gather together a number of classroom items, such as erasers and pencils. You get 20 seconds to try and memorize them all. Then you close your eyes while your partner removes an object. Can you identify the missing object in 10 seconds?