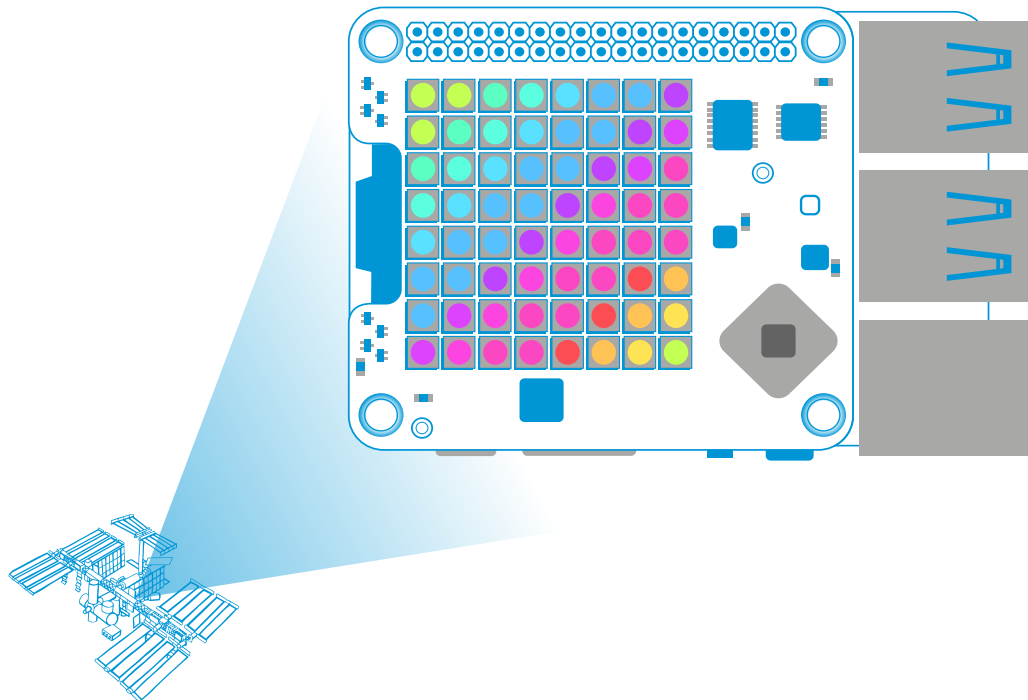
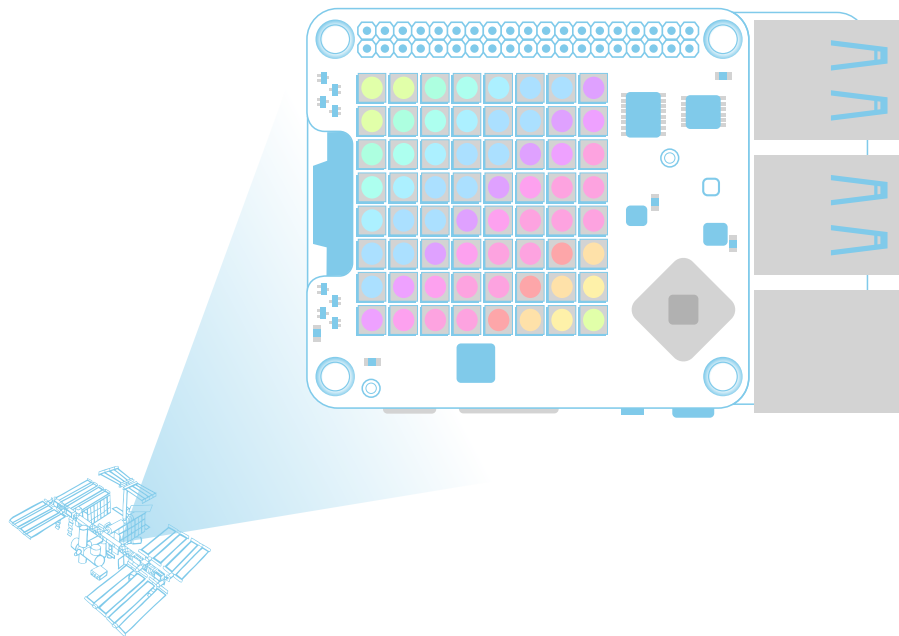


teach with space

→ MEET THE SENSE HAT

Displaying text and images on the Sense HAT LED matrix





Activity 1 – Assemble the Sense HAT

page 4

Activity 2 – Hello, this is Earth!

page 5

Activity 3 – How do colour displays work?

page 6

Activity 4 – Displaying images

page 9

Activity 5 – Setting orientation

page 12

→ MEET THE SENSE HAT

Displaying text and images on the Sense HAT LED matrix

The Sense HAT is an add-on board for Raspberry Pi that was created for the Astro Pi competition. It is a fundamental part of your Astro Pi mission.

The board adds the ability to sense all kinds of information using a LED matrix.

In this set of activities you will explore the Sense HAT hardware and its Python library. You will learn how to control the LED matrix and how to display visual output.

Figure A1



↑ [Astro Pi logo](#)

Equipment

- Astro Pi Kit
- Monitor
- USB keyboard
- USB mouse
- Screwdriver

Fast facts

Age range: 12-16 years old

Complexity: easy

Location: indoor room

Includes use of: Astro Pi Kit; monitor; USB keyboard and USB mouse

Outline

The students will learn how tri-colour LEDs combine to form coloured and white light of a range of intensities. They will control the colour of the LEDs together and individually, using different data structures in Python. These include lists and integer variables. Finally, students will use a range of methods from the Sense HAT library to manipulate text and images on the LED screen.

Students will learn

- How to set LED colour and intensity using RGB values, as well as how to use variables that represent different LED colours
- To display scrolling text on the Sense HAT LED screen, and control various properties of the displayed text, such as colour and scroll speed
- To control the colour of text foreground and background
- To use while true loops to endlessly repeat displayed text
- To set individual pixels using their coordinates and other commands
- To rotate or flip text and images on the LED screen

→ ACTIVITY 1 – ASSEMBLE THE SENSE HAT

The two augmented Raspberry Pi's sent to the International Space Station (ISS) are equipped with a Sense HAT (Hardware Attached on Top) board similar to the one that you have in your kit. The board gives Raspberry Pi the ability to sense all kinds of things, from temperature to movement, and output information using a special display - the 8x8 LED matrix. Onboard the ISS the Raspberry Pi's cannot be connected to a screen, so the matrix is the only real form of visual output that the Raspberry Pi's have.

In this activity you are going to assemble the Sense HAT to start exploring its abilities.

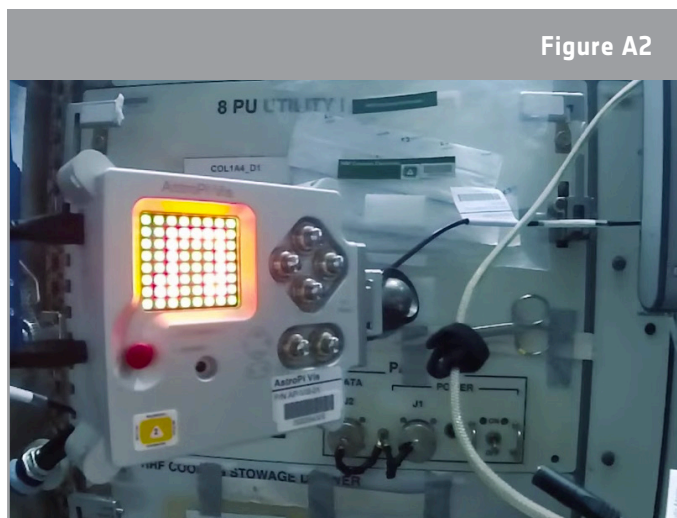


Figure A2

↑ Astro Pi Ed on the ISS

Exercise

1. The Sense HAT comes in a silver anti-static bag, along with the following fixtures and fittings:

- 4 x Hexagon stand-offs (female-to-female)
- 8 x M2.5 screws.

Make sure these are all present before proceeding.

2. You will now attach the Sense HAT to your Raspberry Pi. Do this with Raspberry Pi shut down, disconnected from the mains, and with all other cables disconnected.
3. Screw the hexagon stand-offs to the Raspberry Pi itself, by threading the screws through from the bottom and turning the hexagon stand-offs between finger and thumb (see Figure A3).
4. Next, insert the Sense HAT into the GPIO pin extension header. The corner holes should line up with the hexagon stand-offs.
5. Lastly, put the remaining screws through from the top.

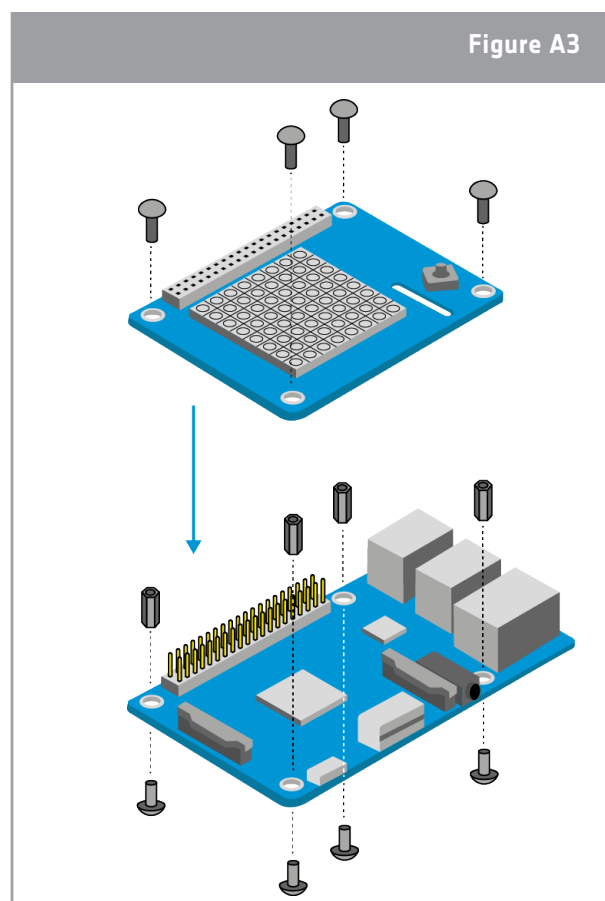


Figure A3

↑ How to assemble the Sense HAT to Raspberry Pi

6. Use a small screwdriver to tighten each corner stand-off individually. They don't need to be especially tight, just enough to ensure that the HAT doesn't become loose.
7. You are now ready to turn on and boot your Astro Pi!

→ ACTIVITY 2 – HELLO, THIS IS EARTH!

The Sense HAT LED matrix is used to display shapes, icons, and messages to the ISS crew. In this activity you are going to run your first program using the Sense HAT and send a message to the astronauts onboard the ISS.

Exercise

1. Open Python 3 by clicking on the Raspberry logo at the top of the screen. This will open the Menu. Select Programming > Python 3. This will cause a Python Shell window to appear. Select File > New File, and type the code below into the new window.

```
File Edit Format Run Options Windows Help
#to allow the program to use the Sense HAT hardware
from sense_hat import SenseHat

#to create a sense object which represents the Sense HAT
sense = SenseHat()

#to make the Sense HAT show the text
sense.show_message("Hello, this is Earth!")
```

2. Select File > Save As and choose a file name for your program, then select Run > Run module. Your message should then scroll across the LED matrix in white text.
3. You can also add a scroll speed parameter. Copy and run the following code:

```
File Edit Format Run Options Windows Help
#to allow the program to use the Sense HAT hardware
from sense_hat import SenseHat

#to create a sense object which represents the Sense HAT
sense = SenseHat()

#to make the Sense HAT show the text
sense.show_message("Hello, this is Earth!", scroll_speed=0.05)
```

4. The LED matrix can also display a single character, rather than an entire message, using the `sense.show_letter` command. Open a new file and type the following code:
5. Save your code and run it. What does it do?

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat

#to allow the program to use the time module
import time

sense = SenseHat()

sense.show_letter("H")
time.sleep(0.5)
sense.show_letter("I")
time.sleep(0.5)

#to reset the LEDs to off
sense.clear()
```

→ ACTIVITY 3 – HOW DO COLOUR DISPLAYS WORK?

The Sense HAT LED matrix contains 64 multi-colour LEDs. Inside each of the 64 LEDs there are three smaller LEDs: a red, a green, and a blue, just like in your TV or smartphone screen.

In this activity you are going to learn how colour displays in electronic systems work, and how to use the colour LEDs to send colourful messages.

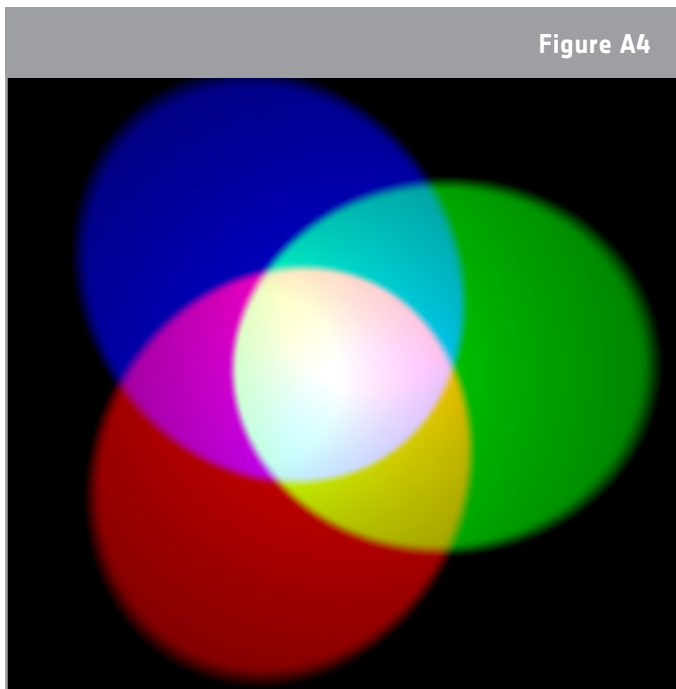
Exercise

1. In additive colour mixing, the colours red, green, and blue are used to make other colours. These are called the three additive primary colours. In Figure A4, there are three spotlights of equal brightness, one for each colour. In the absence of any colour, the result is black.

a) Each combination of two primary colours adds to make a third colour. Write below what colours are made by mixing each combination of primary colours.

b) What is the result of adding all three primary colours?

Figure A4



↑ Additive colour mixing

2. It's possible to make even more colours than the ones you can see in the colour wheel by varying the brightness of the three original colours used. Open a new Python 3 window and write the code below.

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat

#get the sense hat library
sense = SenseHat()

#set r, g and b brightness to maximum
r = 255
g = 255
b = 255

#clear whatever was there before, and then set all pixels to white
sense.clear((r, g, b))
```

Select File > Save As and choose a file name for your program. Then select Run > Run Module. What colour was displayed in the Sense HAT matrix?

3. The variables r, g, and b represent the colours red, green, and blue. The numbers they contain specify how bright each colour should be; brightness can be between 0 and 255. In the code above, the maximum value (255) for each colour was used. Change the values to specify 255 red but 0 green and 0 blue, then run the code again. What other colours can you make?
-
-

4. This colour mixing system is used throughout the Astro Pi programming module. You can use colour mixing to great effect by programming scrolling text. Type the following code into a new file:

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat
sense = SenseHat()

#use show_message function and set text to red
sense.show_message("Hello, this is Earth!", text_colour=(255,0,0))
```

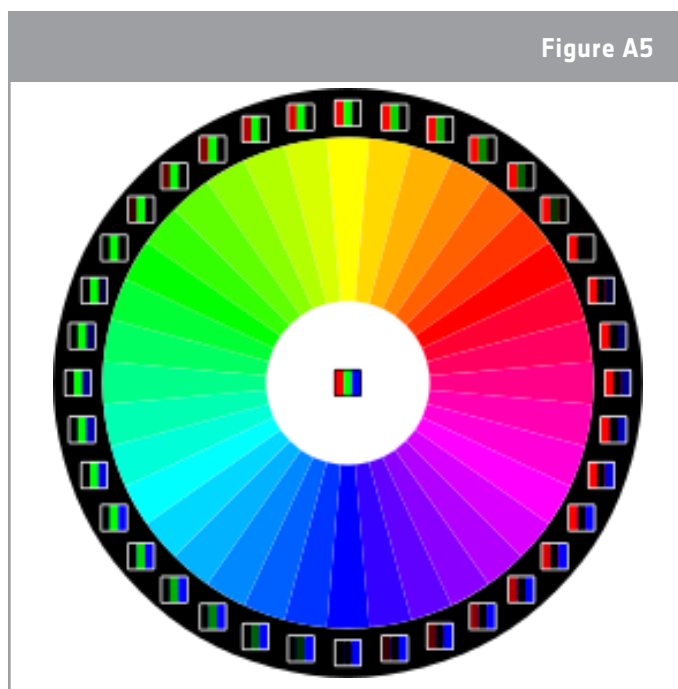
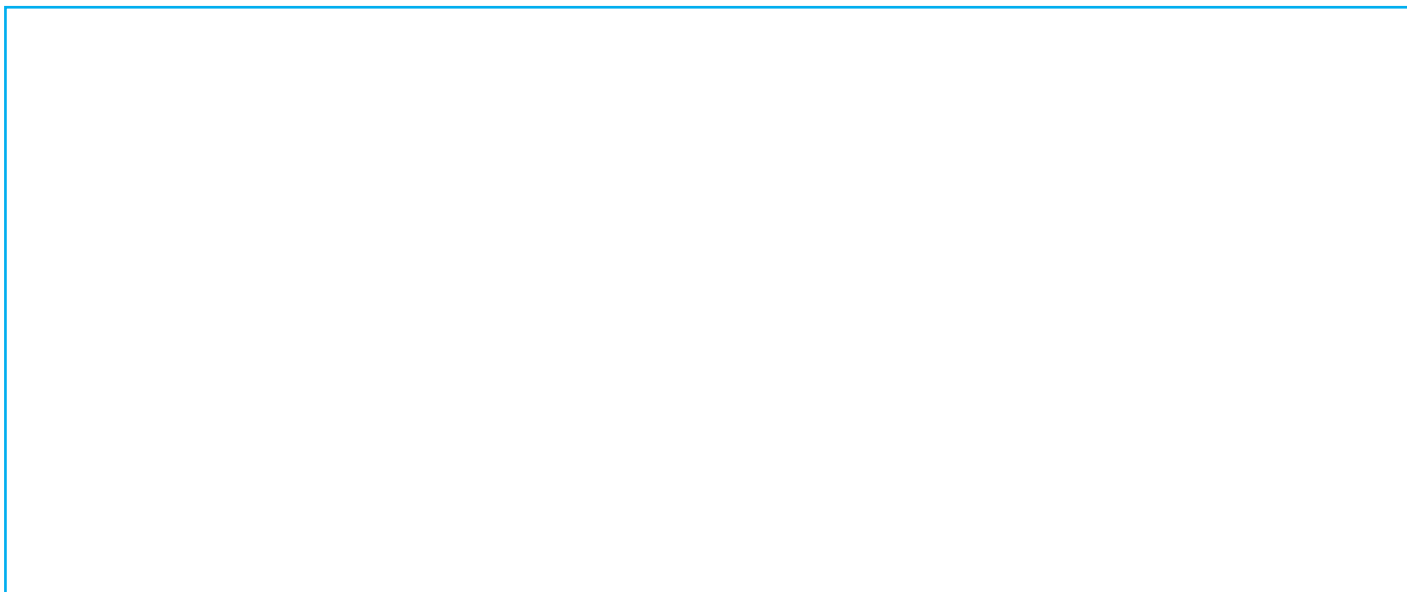
Note: the syntax text_colour=(255, 0, 0) - Don't forget the commas!

5. You can also modify the background colour for the message, like this:

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat
sense = SenseHat()
sense.show_message("Hello, this is Earth!", text_colour=(255,0,0), back_colour=(0,0,255))
sense.clear()
```

Note: The commas are important, don't forget them!

6. Create a message using contrasting colours from opposite sides of the colour wheel, creating an easy-to-read display for busy astronauts. Can you make the message display forever? Copy your code to the box below.



↑ Colour wheel

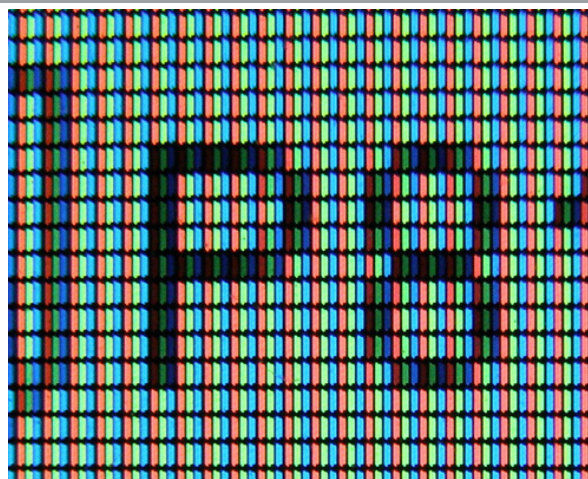
→ ACTIVITY 4 – DISPLAYING IMAGES

The LED matrix can display more than just text. You can also create an image! If you magnify a digital image, you will see thousands of small squares, each filled with a single colour. The combination of these squares – or pixels – forms the image. In this activity you are going to learn what a pixel is and how to display images using the Sense HAT LED matrix.

Exercise

- Figure A6 shows the pixels on a laptop LCD screen. A pixel is a small picture element in a digital image, like the tiny squares you can see on your mobile phone. Each pixel on your screen consists of three sub-pixels (red, green, and blue). You can see that the pixels are turned on and off to form the pattern of letters and numbers. This is how all computer and smartphone screens work. If you want to make recognisable shapes on the LED matrix, this is what you also need to do.

Figure A6



↑ Digital pictures are composed of pixels

Open a new Python 3 window and write the code below.

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat

sense = SenseHat()

sense.clear()

#set X and Y coordinates to 0,0
x = 0
y = 0

#set pixel 0,0 to bright white
sense.set_pixel(x, y, 255, 255, 255)
```

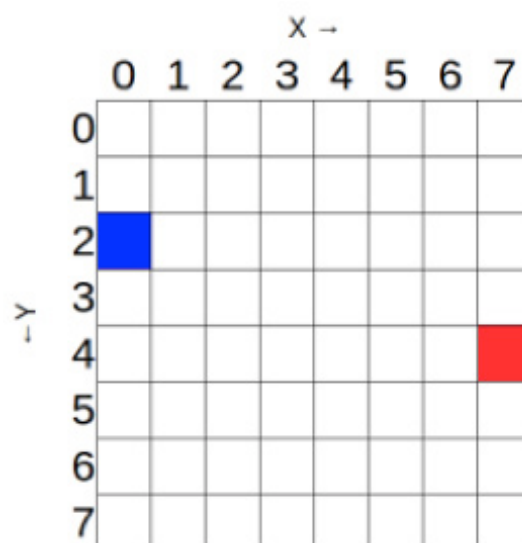
Select File > Save As and choose a file name for your program. Then select Run > Run Module. What did it do?

2. The 'x' and 'y' variables are used to control which individual LED the set_pixel command should change. X is horizontal and ranges from 0 on the left to 7 on the right. Y is vertical and ranges from 0 at the top to 7 at the bottom. Type the following code into a new window:

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat

sense = SenseHat()

sense.set_pixel(0, 0, 255, 0, 0)
sense.set_pixel(0, 7, 0, 255, 0)
sense.set_pixel(7, 0, 0, 0, 255)
sense.set_pixel(7, 7, 255, 0, 255)
```



- a) Colour in the matrix on the right with what you expect will be displayed on the Sense HAT matrix.
- b) Save and run your code. Did it do what you expected?
-
3. You may be tempted to try and draw shapes or patterns using the set_pixel command over and over in your code. There is a set_pixels command though, and with it you can change all 64 LEDs using one line of code! For example, you could draw a Minecraft creeper face on the LED Matrix. And you can use lots of variables to define your colour palette. Run the examples below:

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat

sense = SenseHat()
O= (0, 255, 0)
X= (0, 0, 0)

creeper_pixels = [
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, X, X, 0, 0, X, X, 0,
0, X, X, 0, 0, X, X, 0,
0, 0, 0, X, X, 0, 0, 0,
0, 0, X, X, X, X, 0, 0,
0, 0, X, X, X, X, 0, 0,
0, 0, X, 0, 0, X, 0, 0,
]

sense.set_pixels(creeper_pixels)
```

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat

sense = SenseHat()

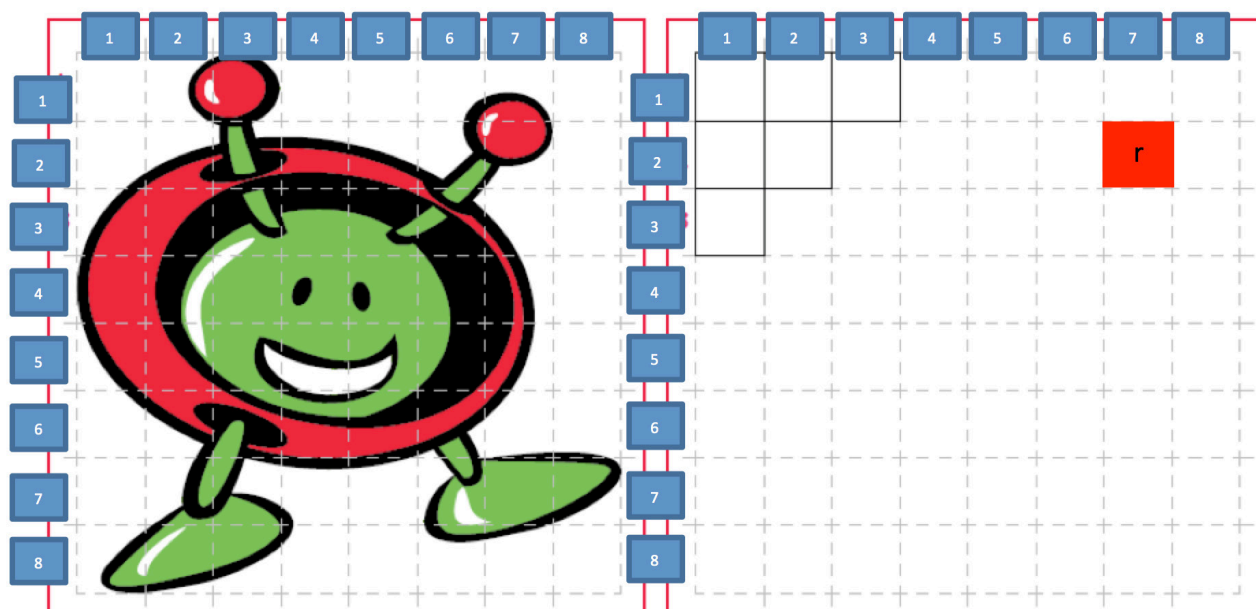
w = [255, 255, 255]
r = [255, 0, 0]
o = [255, 127, 0]
y = [255, 255, 0]
g = [0, 255, 0]
b = [0, 0, 255]
i = [75, 0, 130]
v = [159, 0, 255]
e = [0, 0, 0] # e stands for empty/black

image = [
e,e,e,e,e,e,e,e,
e,e,e,r,r,e,e,e,
e,r,r,o,o,r,r,e,
r,o,o,y,y,o,o,r,
o,y,y,g,g,y,y,o,
y,g,g,b,b,g,g,y,
b,b,b,i,i,b,b,b,
b,i,i,v,v,i,i,b
]

sense.set_pixels(image)
```

Note: the [] and the syntax sense.set_pixels(name of image)!

4. It's time for you to create your own code image. Try to pixel Paxi, ESA's mascot, using the grid below. For each square, the colour should be the main colour in each square that makes up Paxi's image. Write down the code that will allow you to display Paxi's image on the LED matrix.



5. Open a new Python 3 window and run your code. Is Paxi displayed on your LED matrix?

→ ACTIVITY 5 – SETTING ORIENTATION

So far, all your text and images have appeared the same way up, assuming that the HDMI port is at the bottom. However, this may not always be the case (especially in the ISS). In this activity you are going to learn how to change the orientation of the matrix and how to rotate the visual output displayed on it.

Exercise

1. For you to change the orientation of the matrix you can use the `sense.set_rotation()` method, and inside the brackets enter one of four angles (0, 90, 180, 270). Open a new Python 3 window and write the code below. Select File > Save As and choose a file name for your program. Then select Run > Run Module.

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat

sense = SenseHat()

r = [255, 0, 0]
o = [255, 127, 0]
y = [255, 255, 0]
g = [0, 255, 0]
b = [0, 0, 255]
i = [75, 0, 130]
v = [159, 0, 255]
e = [0, 0, 0]

image = [
e,e,e,e,e,e,e,e,
e,e,e,r,r,e,e,e,
e,r,r,o,o,r,r,e,
r,o,o,y,y,o,o,r,
o,y,y,g,g,y,y,o,
y,g,g,b,b,g,g,y,
b,b,b,i,i,b,b,b,
b,i,i,v,v,i,i,b
]

sense.set_pixels(image)
sense.set_rotation(180)
```

2. You can also create spinning text using a "for" loop. Open a new Python 3 window. Type and run the code on the right. What did it do?

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat
from time import sleep

sense = SenseHat()

sense.show_letter("J")

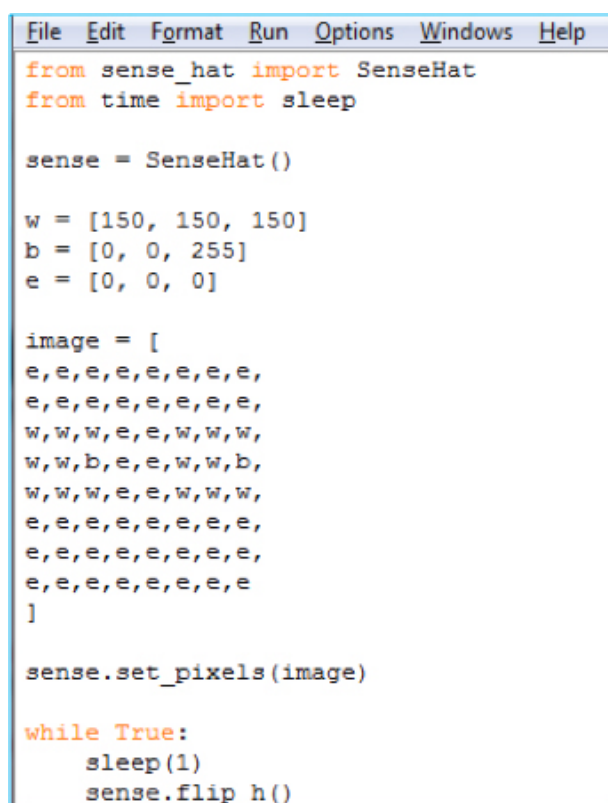
#create a list of angles to iterate through
angles = [0, 90, 180, 270, 0, 90, 180, 270]

#cycles through the angles, rotating
for r in angles:
    sense.set_rotation(r)
    sleep(0.5)
```

3. You can also flip the image on the screen, either horizontally or vertically, using these code lines:

```
sense.flip_h()
or
sense.flip_v()
```

With the example below you can create a simple animation by flipping the image repeatedly:



```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat
from time import sleep

sense = SenseHat()

w = [150, 150, 150]
b = [0, 0, 255]
e = [0, 0, 0]

image = [
e,e,e,e,e,e,e,e,
e,e,e,e,e,e,e,e,
w,w,w,e,e,w,w,w,
w,w,b,e,e,w,w,b,
w,w,w,e,e,w,w,w,
e,e,e,e,e,e,e,e,
e,e,e,e,e,e,e,e,
e,e,e,e,e,e,e,e
]

sense.set_pixels(image)

while True:
    sleep(1)
    sense.flip_h()
```

EXTENSION

Can you create a spinning image using one of the drawing techniques already shown, and use the `sense.set_rotation` method to make it spin?

NEXT STEPS

Using output devices like the Sense HAT matrix is a fantastic way for you to display text and images using your Astro Pi. The Raspberry Pi Sense HAT also contains a whole set of sensors that can be used to sense the space around you and to do amazing science experiments - in your classroom and onboard the ISS! In the next set of activities you will explore the possible things you can do with Sense HAT sensors.

teach with space – meet the sense hat | T05.2b
www.esa.int/education

An ESA Education production
in collaboration with Raspberry Pi Foundation, ESERO Poland and ESERO UK

Copyright 2017 © European Space Agency