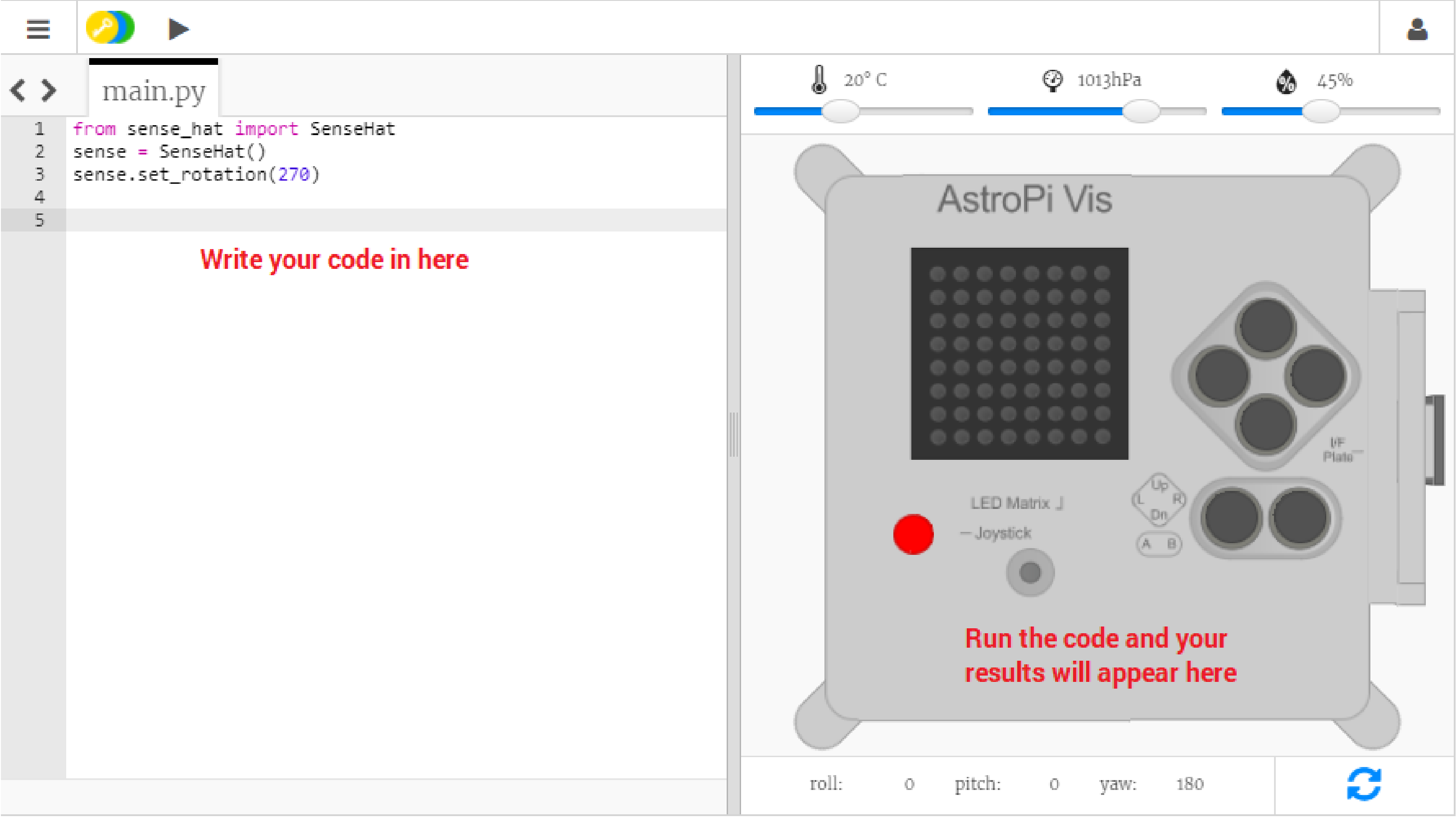
Bodgers

Last Week

We used the online Sense HAT emulator to create our program, https://trinket.io/mission-zero

We followed the tutorial at https://projects.raspberrypi.org/en/projects/astro-pi-mission-zero



```
from sense_hat import SenseHat
sense = SenseHat()
sense.set_rotation(270)
```

Write your code in here

Run the code and your results will appear here

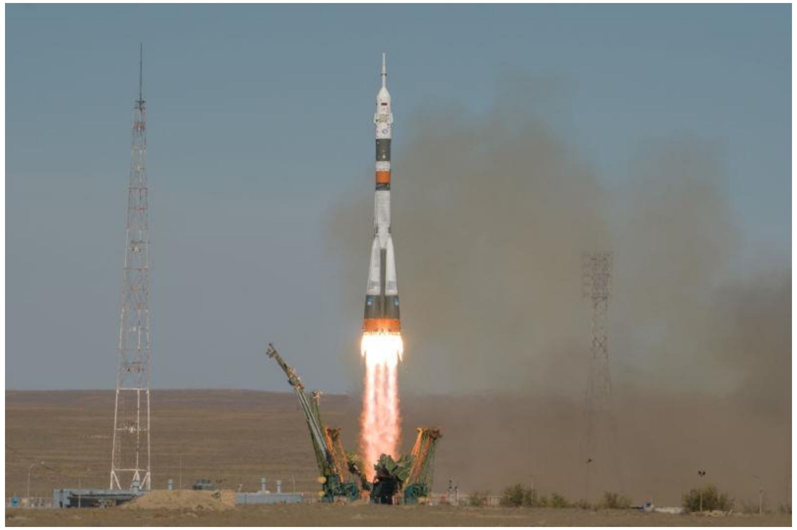
Our Trinket: https://trinket.io/python/04b90b70cf

This week:

1. Look at Thursday’s Soyuz incident
2. Finish our messages

What Happened?

- Last Thursday in Kazakhstan, a Soyuz rocket took off carrying NASA astronaut Nick Hague and Russian cosmonaut Aleksey Ovchinin to the International Space Station.
- Everything went as planned until the separation of the rocket's final stage.Because the rocket not reached orbit at this point. The crew was forced to make a rapid ballistic descent.
- After about 20 minutes of uncertainty, Russian officials confirmed the crew were ok and had landed about 20km east of Dzhezkazgan, in central Kazakhstan.



Soyuz rocket failure

AFP

What Does This Mean For The ISS

- Three people remain on the station: American astronaut Serena Auñón-Chancellor, German ISS Commander Alexander Gerst, and Russian Sergey Prokopyev. They were due to return to Earth in mid-December.
- The crew has consumables for months and could be re-supplied by two US cargo ships as well as a Japanese vehicle.
- The bigger issue is the expiry date of the Soyuz spacecraft docked to the station. Generally, these vehicles are rated to survive about 200 days on orbit, and this ends in January.
- The ISS can be operated without a crew, but if anything goes wrong there would be no one there to repair it.

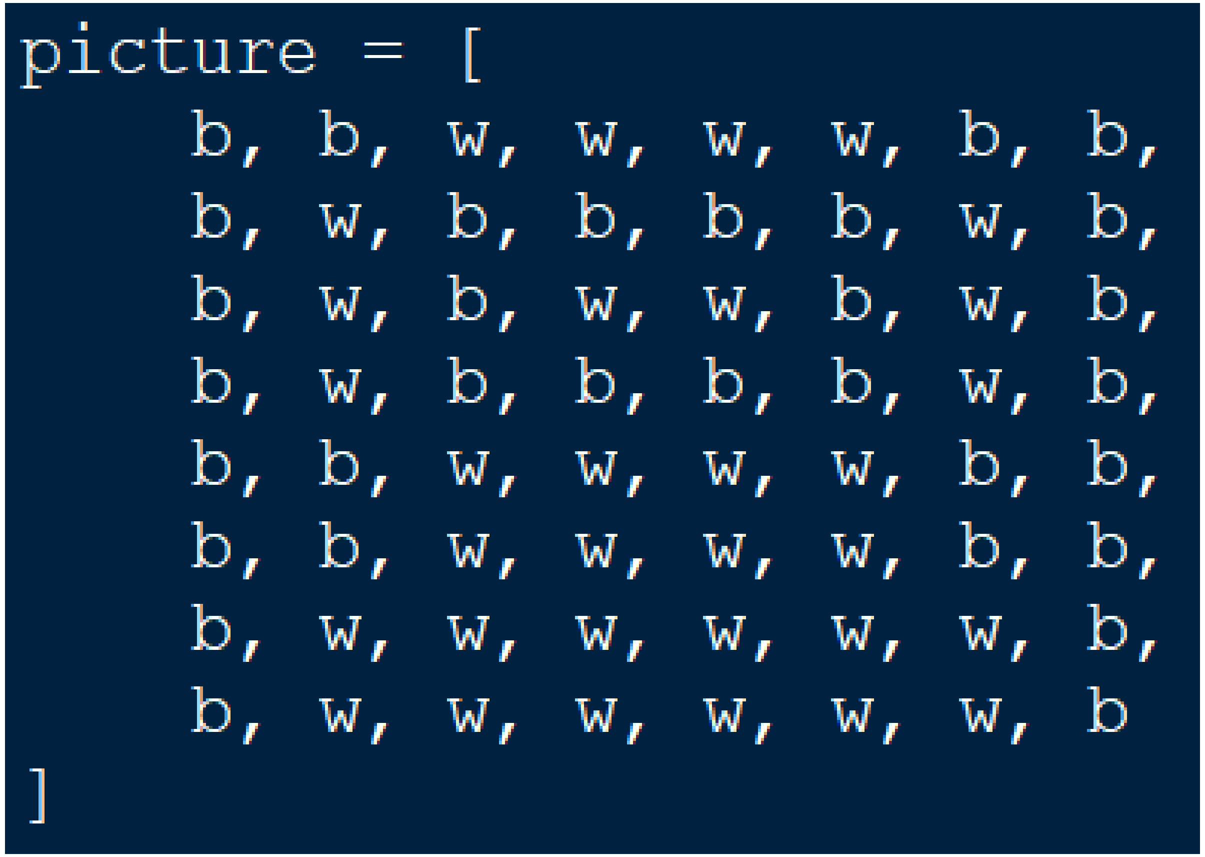
What Happens Next?

- The Soyuz is grounded until what caused the problem is identified and fixed.
- Three astronauts—American Anne McClain, Canadian David Saint-Jacques, and Russian Oleg Kononenko—were scheduled to launch to the station in mid-December aboard a Soyuz vehicle.
- If the Russians and NASA aren't fully confident that the problem is fixed, a Soyuz spacecraft could possibly be launched empty to the station and replace the existing Soyuz as a lifeboat for the three astronauts on orbit. This would give the on-orbit crew another 200 days.

Our Trinket: https://trinket.io/python/04b90b70cf

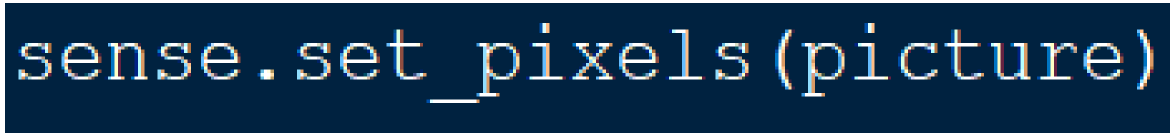
Below your new variables, create a list of 64 items. Each item represents one pixel on the LED matrix, and corresponds to one of the colour variables you defined.

Draw your picture by putting a variable where you want its assigned colour to appear. We have drawn an astronaut by using the black (b) pixels as the background and the white (w) pixels to draw the astronaut’s space suit:



```
picture = [
    b, b, w, w, w, w, b, b,
    b, w, b, b, b, b, w, b,
    b, w, b, w, w, b, w, b,
    b, w, b, b, b, b, w, b,
    b, b, w, w, w, w, b, b,
    b, b, w, w, w, w, b, b,
    b, w, w, w, w, w, w, b,
    b, w, w, w, w, w, w, b
]
```

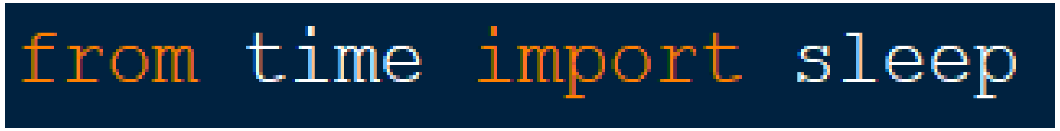
Add a line of code to display your picture on the LED display.



```
sense.set_pixels(picture)
```

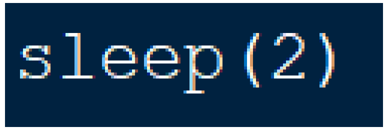
Press Run to see your picture displayed.

You might want to add some code to include a short wait (or sleep) after the picture is displayed. This will give the astronauts time to see your picture before the next part of your message appears. At the top of your program, add:



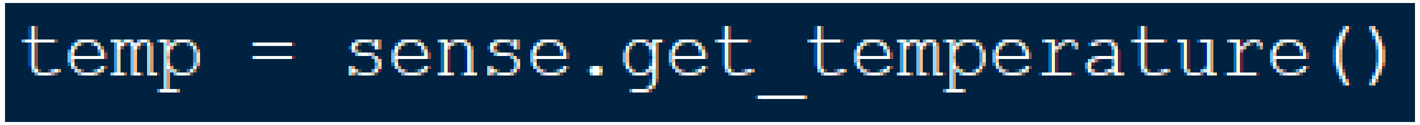
```
from time import sleep
```

Then, on the line after the one that displays your picture, add this code to wait for two seconds:



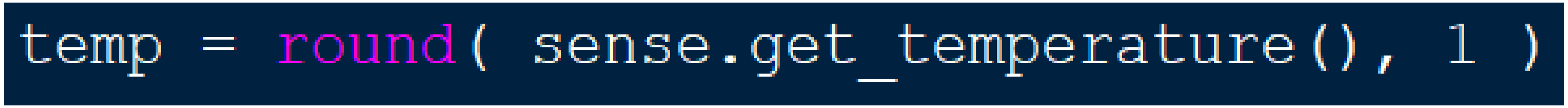
```
sleep(2)
```

Add this code to take a temperature reading:



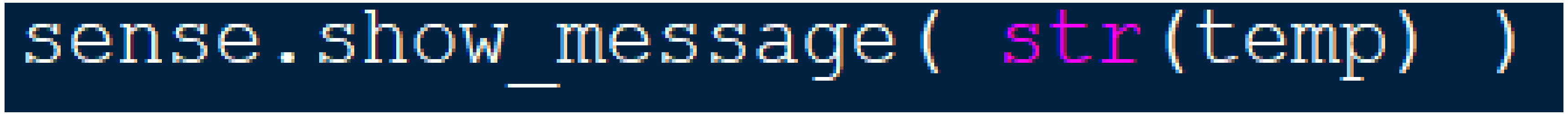
```
temp = sense.get_temperature()
```

The temperature is recorded very precisely, i.e. the stored value will have a large number of decimal places. You can round the value to any number of decimal places.



```
temp = round( sense.get_temperature(), 1 )
```

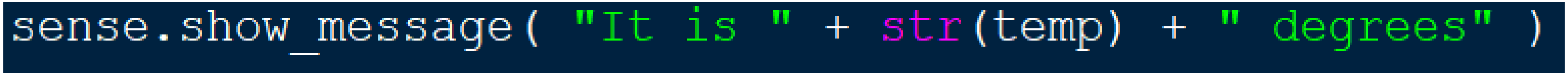
To display the current temperature as a scrolling message on the display, add this line of code:



```
sense.show_message( str(temp) )
```

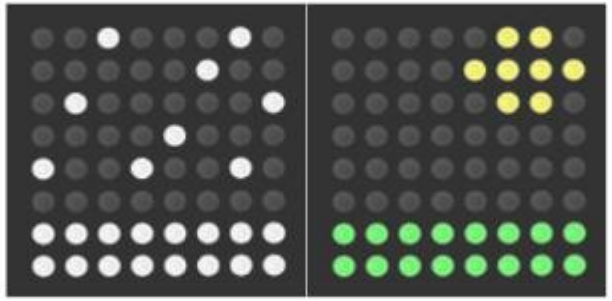
The str() part converts the temperature from a number into text so that the Astro Pi can display it.

You can also display the temperature as part of another message by joining the parts of your message together with a +.



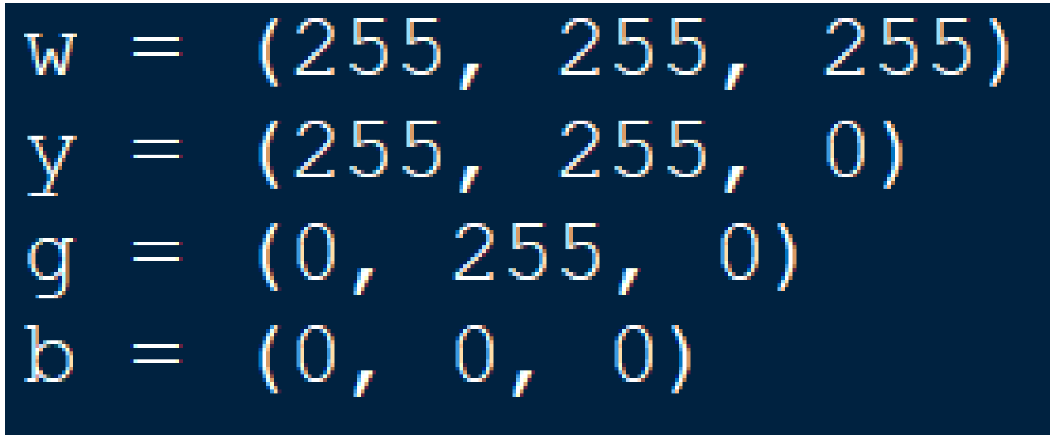
```
sense.show_message( "It is " + str(temp) + " degrees" )
```

Display The Temperature



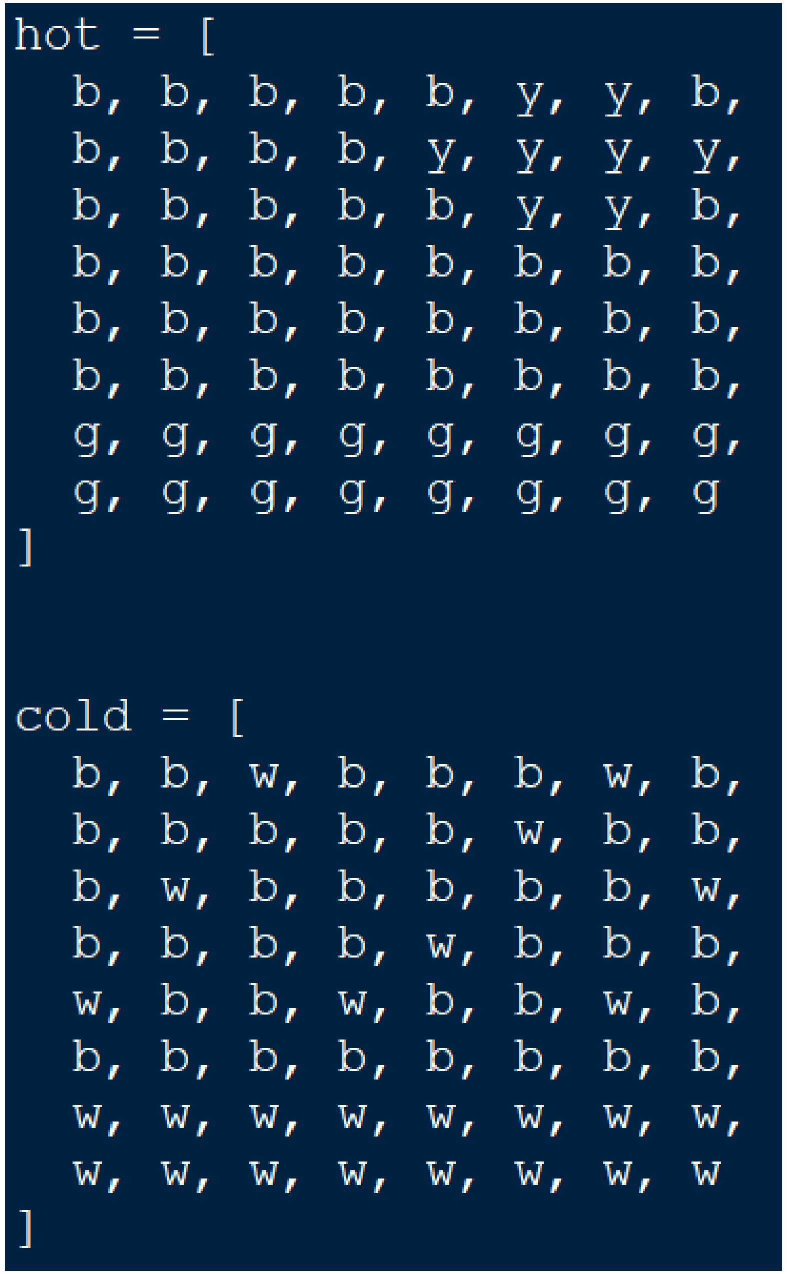
You could combine your temperature reading with a picture to show the temperature. For example, you might display a snowstorm for cold temperatures, and a sunny day for hot temperatures.

At the bottom of your program, create more colour variables for any colours you want to use in your pictures. You may already have defined some of them in a previous step. In our examples we will use white (w), yellow (y), green (g), and black/blank (b).



```
w = (255, 255, 255)
y = (255, 255, 0)
g = (0, 255, 0)
b = (0, 0, 0)
```

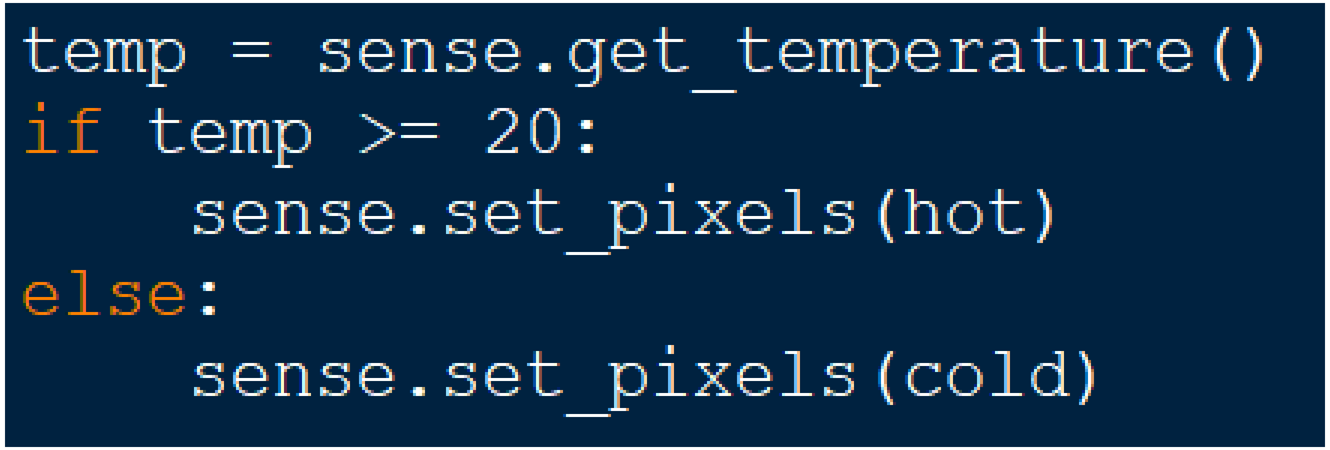
Just like earlier, draw your pictures by first creating a list for each of them, and then setting the list items to the colours you want your pixels to be.



```
hot = [
  b, b, b, b, b, y, y, b,
  b, b, b, b, y, y, y, y,
  b, b, b, b, b, y, y, b,
  b, b, b, b, b, b, b, b,
  b, b, b, b, b, b, b, b,
  b, b, b, b, b, b, b, b,
  g, g, g, g, g, g, g, g,
  g, g, g, g, g, g, g, g
]


cold = [
  b, b, w, b, b, b, w, b,
  b, b, b, b, b, w, b, b,
  b, w, b, b, b, b, b, w,
  b, b, b, b, w, b, b, b,
  w, b, b, w, b, b, w, b,
  b, b, b, b, b, b, b, b,
  w, w, w, w, w, w, w, w,
  w, w, w, w, w, w, w, w
]
```

Now decide which picture to display. For this example, we will display the hot image if the temperature reading is 20 degrees or above, and the cold image if the temperature is below 20 degrees.



```
temp = sense.get_temperature()
if temp >= 20:
    sense.set_pixels(hot)
else:
    sense.set_pixels(cold)
```

Use the temperature slider to set a temperature on the emulator. Run your program and check that the image you've selected for that temperature is correctly displayed.

That's It For Now

As we can't enter until the end of October we will leave the Astro Pi for now and we will come back in a few weeks to finalize our entries.
Next we will look at Pygame Zero where after I do a demo we'll have a go at designing our own game.