

Fire Danger Warning Sign

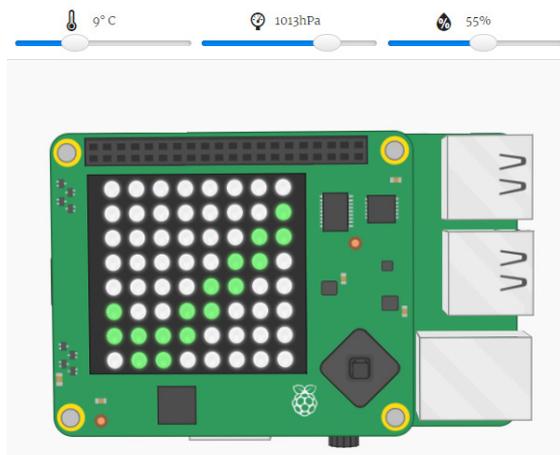
Create your own warning sign to prepare for bushfire season.



INTRODUCTION

What you will make

You will create an early warning system that senses temperature and humidity using a SenseHat and a Raspberry Pi Emulator.



What you will learn

- Read data from an external sensor.
- Make decisions based on data.
- Generate images using a grid of pixels.

Code Club Australia recognises the Traditional Custodians of the land across Australia and their continuing connection to land, cultures, and communities. Australia's traditional owners are the world's first innovators.

What you will need

HARDWARE

A computer capable of accessing Trinket online. You do not need a Raspberry Pi to complete this project.

DOWNLOADS

Offline starter project
bit.ly/mhwarningsign

Additional notes for educators

Here is a link to the completed project
<https://trinket.io/python/d4da2fd69f>

Check out our blog post for this project with tips, curriculum and supporting material at
medium.com/@codeclubau

1. USING YOUR SENSEHAT

The first thing we need to do is set up our SenseHat so it's ready to go.

- Open the starter project at bit.ly/mhwarningsign
- If you have a physical Raspberry Pi with a SenseHat, open a new Python project in Thonny.
- The first line will tell Python to use version 3 of Python.
- If you're using the online emulator, this should already be there for you.



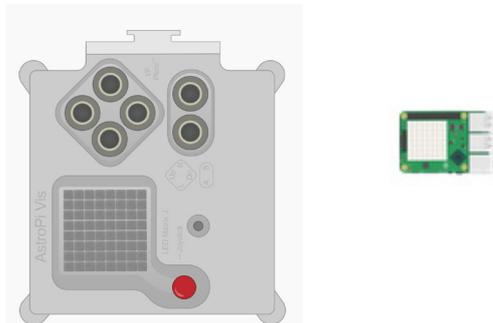
```
1 #!/bin/python3
2
3 |
```

- We want to tell Python that we will be using the SenseHat, so the first thing we need to do is to import the SenseHat module.
- Add this code, then press Run at the top of the screen to test it.

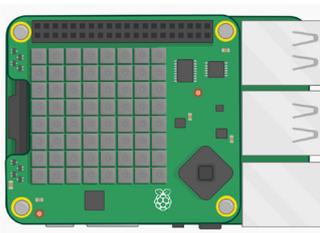


```
1 #!/bin/python3
2
3 from sense_hat import SenseHat
```

- You might see an AstroPi appear on the right of the screen.
- We want to use the regular Raspberry Pi, not the AstroPi, so press the little Raspberry Pi toggle button on the bottom right of the screen.



- The AstroPi should turn into the regular Raspberry Pi.



- Finally, for this step, we're going to use a variable to refer to our SenseHat to make our code easier.
- Add the following line of code.



```
1 #!/bin/python3
2
3 from sense_hat import SenseHat
4
5 sense = SenseHat()
```

2. USING THE SENSORS

The SenseHat is loaded with a bunch of sensors that can detect all sorts of things, including temperature, pressure, humidity, acceleration and magnetic fields. In this project, we're going to use the temperature and humidity sensors to determine fire risk.

- First, we'll assign the temperature and humidity measured by the SenseHat to variables so that we can use them later.



```
5 sense = SenseHat()  
6  
7 temperature = sense.temperature  
8 humidity = sense.humidity
```

- We should check to make sure that we're getting the right values.
- Let's print out the values for temperature and humidity.



```
7 temperature = sense.temperature  
8 humidity = sense.humidity  
9 print(temperature, humidity)
```

- Run your program.
- A console should appear at the bottom right of your window that will output the results of the program. The first number is the temperature and the second is the humidity.



```
Powered by  trinket  
20.0131443421 45.0865492074
```

- Change the temperature and humidity sliders.



- Run your program again. You should get the new reading.



```
Powered by  trinket  
39.0227514106 64.4465261636
```

- It's a bit of a pain to have to run the program every time we want to get a new reading. Let's put it in a loop to constantly get new readings instead.



```
5 sense = SenseHat()  
6  
7 while True:  
8     temperature = sense.temperature  
9     humidity = sense.humidity  
10    print(temperature, humidity)
```

- Note: the lines that are being looped through (lines 8-10) have been shifted forward by two spaces. This is called indenting and it tells python that these lines belong inside the loop. You can indent these lines by highlighting them all (lines 8-10) and pressing the Tab key.

3. CHECK THE FIRE DANGER

The real fire danger, measured by experts, has a lot of factors. Two important factors are temperature and humidity. We're going to measure these and if we think there might be a risk, we're going to prompt the user to check the fire danger with their state or territory fire service.

- We need to check if the temperature is above 20 degrees, or the humidity is less than 50%. If it is, we will warn the user.
- Test your code here. If you're getting an error it may be that your indentation is wrong.

```
7 while True:
8     temperature = sense.temperature
9     humidity = sense.humidity
10    print(temperature, humidity)
11    if temperature > 20 or humidity < 50:
12        sense.show_message("Danger")
```

- These are conservative figures, we will show a lot of false positives, meaning that we will warn the user unnecessarily. In some situations, like this one, it's better to have lots of false positives, than risk false negatives (when there's actually danger, but we don't tell them). In this case, all we're doing is encouraging the user to check the fire danger with their state or territory fire service.

- Let's add an else statement that will tell the user when there's a low risk.
- Run your program. The words "Danger" or "Low risk" should scroll across the screen, depending on the temperature and humidity.
- If you have an error, make sure you check your indentation.

```
7 while True:
8     temperature = sense.temperature
9     humidity = sense.humidity
10    print(temperature, humidity)
11    if temperature > 20 or humidity < 50:
12        sense.show_message("Danger")
13    else:
14        sense.show_message("Low risk")
```

4. VISUAL ICON

Instead of displaying a message in text, it can sometimes be better to display an image to the user. This displays a clear message without the user having to wait for the message to scroll across the screen. We will use two icons: one for danger, and one for low risk.

- To show an image on the SenseHat, we first need to define our colours. Starting with the low risk image, we will need two colours: green and white, which we will represent as the letters G and W.

```
1  #!/bin/python3
2
3  from sense_hat import SenseHat
4
5  sense = SenseHat()
6
7  G = [0, 255, 0] # Green
8  W = [255, 255, 255] # White
9
10 while True:
11     temperature = sense.temperature
12     humidity = sense.humidity
```

- We are using an array of three numbers to represent the colours. These three numbers are the amount of Red, the amount of Green, and the amount of Blue. This is often called the RGB value of a colour. We need to add this definition at the beginning of our code. You can press Enter to add new lines.
- Any writing that comes after the # is a comment on the code. This does not affect the coding, but may be useful for you to remember what the colours are.

- Now that we have our two colours, we can create our image. The SenseHat has 64 LED pixels, each of which need to be assigned a colour to create our image. We're going to make a green tick on a white background.
- To save you from having to type this out, the array has been provided in snippets.py, (which you can find in a separate tab near the top of the screen). Don't forget to include the brackets when you copy+paste.

```
7  G = [0, 255, 0] # Green
8  W = [255, 255, 255] # White
9
10 lowRisk = [
11     W, W, W, W, W, W, W, W,
12     W, W, W, W, W, W, W, G,
13     W, W, W, W, W, W, G, G,
14     W, W, W, W, W, G, G, W,
15     W, W, W, W, G, G, W, W,
16     G, W, W, G, G, W, W, W,
17     G, G, G, G, W, W, W, W,
18     W, G, G, W, W, W, W, W
19 ]
```

- Next, we need to make an icon for Danger. This one will use Red, Black, and Yellow (as well as White, which we've already defined).

```

5  sense = SenseHat()
6
7  G = [0, 255, 0] # Green
8  W = [255, 255, 255] # White
9  R = [255, 0, 0] # Red
10 B = [0, 0, 0] # Black
11 Y = [255, 255, 0] # Yellow
12
13 lowRisk = [
14     W, W, W, W, W, W, W, W,
15     W, W, W, W, W, W, W, G,
16     W, W, W, W, W, W, G, G,

```

- Now we can draw the icon for danger. This one is also on the snippets.py tab.

```

19     G, W, W, G, G, W, W, W,
20     G, G, G, G, W, W, W, W,
21     W, G, G, W, W, W, W, W
22 ]
23
24 danger = [
25     W, W, W, B, B, W, W, W,
26     W, W, B, Y, Y, B, W, W,
27     W, B, Y, R, Y, Y, B, W,
28     B, Y, Y, R, Y, Y, Y, B,
29     B, Y, Y, Y, Y, Y, Y, B,
30     W, B, Y, R, Y, Y, B, W,
31     W, W, B, Y, Y, B, W, W,
32     W, W, W, B, B, W, W, W
33 ]
34
35 while True:
36     temperature = sense.temperature
37     humidity = sense.humidity
38     print(temperature, humidity)

```

- Now our hard work is about to pay off! Let's change our show_message statements to instead show the icons using set_pixels.

```

39 if temperature > 20 or humidity < 50:
40     sense.set_pixels(danger)
41 else:
42     sense.set_pixels(lowRisk)

```

- Run your program. It should now show the danger sign when the temperature is too high, or the humidity is too low. Otherwise, it should show the low risk sign.

Challenges:

Your own icons

- We've provided you with the code to create the icons, but can you change the icons provided?
- Hint: You can find the RGB values for more colours at jumpto.cc/colours

Advanced Challenge: Hardware

- Do you have access to a Raspberry Pi and a SenseHAT? Your local Library may have one.
- Can you build this program on the Raspberry Pi? How useful is it as a real-world warning system.
- How could you improve its accuracy?

Advanced Challenge: Other seasons

- Currently, we're using temperature and humidity to provide an early warning for fire danger, but these values can predict other things too. What other dangerous weather conditions might people need to know about? Can you add another step in your if statement to warn people? Think about what conditions are needed and what that warning might look like?
- Hint: You will need to use an elif statement as part of your branching logic.

Congratulations you're a
Moonhack changemaker!

Don't forget to talk to an adult
about registering your
participation at
moonhack.com

