

The starter code

Your printed copy of `image_starter.py`.

Name: _____

Date: _____

Set up: keep `image_starter.py`, `cow-demo.jpg` and the `cow_message` folder in the *same folder*.
Open `image_starter.py` in Spyder and press **F5**.

```
7 import numpy as np
8 from PIL import Image
9 import matplotlib.pyplot as plt
10 ALPHABET = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"      # A is 0, B is 1, ... Z is 25
11
12
13 # ----- GIVEN: you do not need to change these two functions -----
14 def image_path(number):
15     # image_path(1) gives "cow_message/01.png", image_path(12) gives
16     # "cow_message/12.png"
17     return "cow_message/" + str(number).zfill(2) + ".png"
18
19 def show_colour(red, green, blue):
20     # draws one big square of the colour you give it (each number is 0 to 255)
21     plt.imshow([[red, green, blue]])
22     plt.axis("off")
23     plt.show()
24
25
26 # ----- STEP 1: look at a picture (run this and read the output) -----
27 img = Image.open("cow-demo.jpg")
28 pixels = np.array(img)
29 print("size:", img.size)
30 print("shape:", pixels.shape)
31 print("first pixel:", pixels[0, 0])
32 show_colour(230, 30, 30)      # try changing these three numbers!
33
34
35 # ----- STEP 2 (TODO): add up all the red in ONE picture -----
36 def total_red(path):
37     img = Image.open(path)
38     pixels = np.array(img)
39
40     # TODO: get just the red channel
41     # hint: pixels[:, :, 0] (every row, every column, channel 0 = red)
42     # TODO: add up every red value in it
43     # hint: .sum()
44     # TODO: return that total
45
46     return 0
47
48
49 # Test it. It should print 10218857
50 print(total_red("cow-demo.jpg"))
```

```

53 # ----- STEP 3 (TODO): turn a picture into ONE letter -----
54 def letter_for(path):
55     # TODO: 1. get the red total for this picture (you wrote a function for this!)
56     #         2. take it % 26    (the remainder, a number from 0 to 25)
57     #         3. use that number to pick a letter from ALPHABET
58     #             hint: ALPHABET[2] is "C"
59     #         4. return the letter
60     return ""
61
62
63 # Test it. It should print Z
64 print(letter_for("cow-demo.jpg"))

```

```

67 # ----- STEP 4 (TODO): decode the WHOLE message -----
68 def decode_message():
69     message = ""          # the message we are building. It starts empty.
70
71     # TODO: for each number from 1 to 37 (hint: for number in range(1, 38))
72     #     1. find the picture:  image_path(number)
73     #     2. work out its letter (you already wrote a function for this!)
74     #     3. stick the letter on the end of message:  message += letter
75
76     return message
77
78
79 # Run it. Does the message make sense? If not, check your steps 2 and 3.
80 # print(decode_message())

```

```

83 # ----- STRETCH: a different channel -----
84 # What if you add up the GREEN values (channel 1) or BLUE values (channel 2) instead?
85 # Copy your functions, change the 0, and see what message you get.

```

Tick when done: ☐ colour square shows ☐ total_red prints 10218857 ☐ letter_for prints Z
☐ decode_message works