

The explore code

Your printed copy of `image_explore.py`. Type it or open the file.

Name: _____

Date: _____

Set up: keep `image_explore.py` and `cow-demo.jpg` in the *same folder*. Open the file in Spyder and press **F5** to run everything, or select a block and press **F9**.

```
8 import numpy as np
9 from PIL import Image
10 import matplotlib.pyplot as plt
11
12 # ----- GIVEN: you do not need to change these three functions -----
13 def show(pixels):
14     # draws a colour picture from a grid of numbers
15     plt.imshow(pixels)
16     plt.axis("off")
17     plt.show()
18
19 def show_channel(channel, colour):
20     # draws ONE channel, using the colour scale you give: "Reds", "Greens", "Blues" or
    "gray"
21     plt.imshow(channel, cmap=colour, vmin=0, vmax=255)
22     plt.axis("off")
23     plt.colorbar()
24     plt.show()
25
26 def show_colour(red, green, blue):
27     # draws one big square of the colour you give it (each number is 0 to 255)
28     plt.imshow([[red, green, blue]])
29     plt.axis("off")
30     plt.show()
31
32 # ----- TASK 1: open a picture -----
33 img = Image.open("cow-demo.jpg")
34 pixels = np.array(img)
35 show(pixels)
36 print(pixels.shape)          # predict first: how many rows? how many columns?
37
38 # ----- TASK 2: one pixel -----
39 print(pixels[0, 0])          # the pixel in row 0, column 0 (top left)
40 # TODO: print the pixel in row 100, column 200
41 #   hint: pixels[row, column]
42
43 # ----- TASK 3: crop a corner -----
44 corner = pixels[0:100, 0:100]    # rows 0 to 99, columns 0 to 99
45 show(corner)
46 # TODO: crop a different 100 x 100 square, then show it
47 #   hint: change the numbers, keep the colon and the comma
48
49 # ----- TRY IT: mix a colour (after the RGB slides) -----
50 show_colour(230, 30, 30)        # red, green, blue. Try changing these three numbers!
51 # TODO: make yellow, then make a grey
```

```

61 # ----- TASK 4: split the channels -----
62 red = pixels[:, :, 0]
63 show_channel(red, "Reds")
64 # TODO: do the same for green (channel 1) and blue (channel 2)
65 # hint: show_channel(green, "Greens")
66 # Which channel is brightest on the wall? Which is brightest on the cow?
67
68 # ----- TASK 5: switch a channel off -----
69 no_red = pixels.copy() # .copy() makes a second picture to change
70 no_red[:, :, 0] = 0 # set every red value to 0
71 show(no_red)
72 # TODO: make a picture with NO green, and one with NO blue
73 # TODO: make a picture that is ONLY blue (red and green both 0)
74
75 # ----- TASK 6: grey and negative -----
76 grey = np.array(img.convert("L")) # "L" means one brightness number per pixel
77 show_channel(grey, "gray")
78 negative = 255 - pixels # flips every number: 0 becomes 255, 255 becomes 0
79 show(negative)
80
81 # ----- TASK 7: count and average -----
82 print(grey.mean()) # the average brightness, 0 to 255
83 print((grey < 50).sum()) # how many pixels are darker than 50?
84 # TODO: how many pixels are brighter than 100?
85 # hint: change the < and the number
86 # TODO: what fraction of the whole picture is that?
87 # hint: divide by grey.size
88
89 # ----- TASK 8: a histogram -----
90 plt.hist(grey.flatten(), bins=16)
91 plt.show()
92 # TODO: draw a histogram of just the red channel: plt.hist(red.flatten(), bins=16)
93
94 # ----- STRETCH -----
95 # Mirror the picture: show(pixels[:, ::-1]) # ::-1 means "go backwards"
96 # Upside down: show(pixels[::-1])
97 # Brighten it: add 60 to every number. Careful, what happens above 255?

```

Tick when done:

- ☐ Task 1 ☐ Task 2 ☐ Task 3 ☐ Try it: mix a colour
☐ Task 4 ☐ Task 5 ☐ Task 6 ☐ Task 7 ☐ Task 8