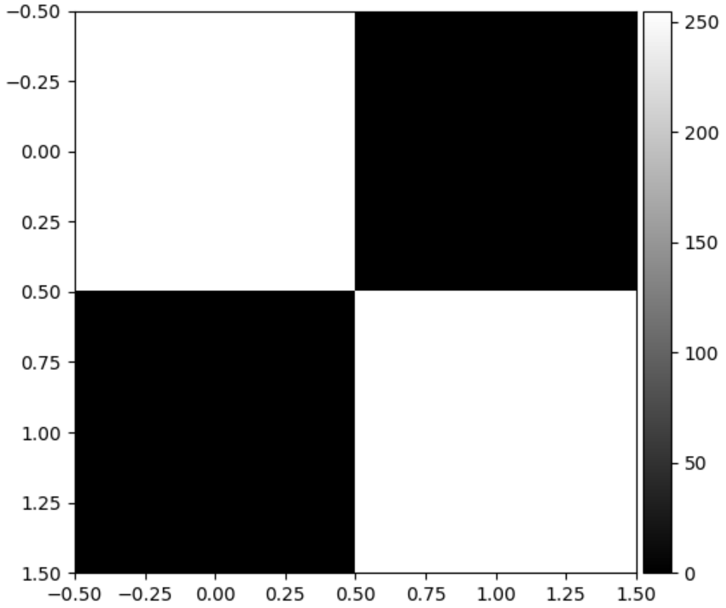
```
# First import the required Python Libraries
import numpy as np
import matplotlib.pyplot as plt
from skimage import img_as_uint
from skimage.io import imshow, imread
from skimage.color import rgb2hsv
from skimage.color import rgb2gray
```

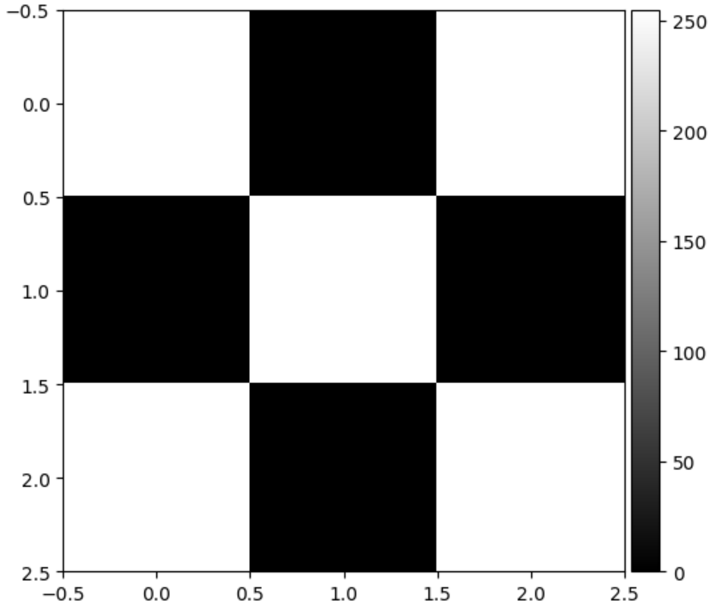
An image can be thought of as a matrix where every pixel's color is represented by a number on a scale.

```
array_1 = np.array([[255, 0],
                    [0, 255]])
imshow(array_1, cmap = 'gray');
```

```
/usr/local/lib/python3.10/dist-packages/skimage/io/_plugins/matplotlib_plugin.py:150:
  lo, hi, cmap = _get_display_range(image)
```



```
array_2 = np.array([[255, 0, 255],
                    [0, 255, 0],
                    [255, 0, 255]])
imshow(array_2, cmap = 'gray');
```

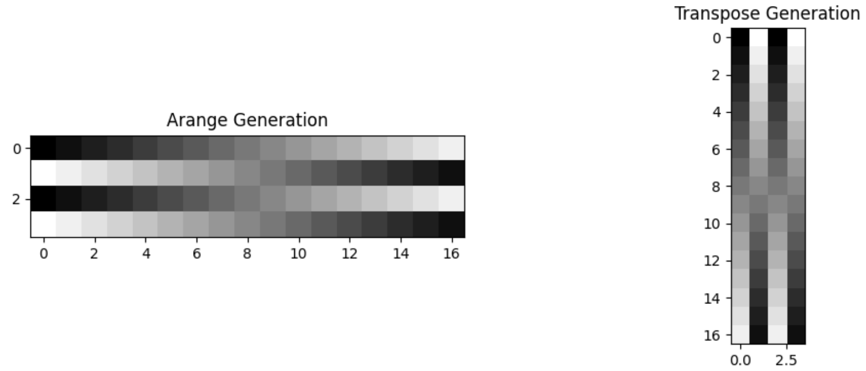


Although our examples were picking colors on the extreme end of the spectrum, we can also access any of the colors in between.

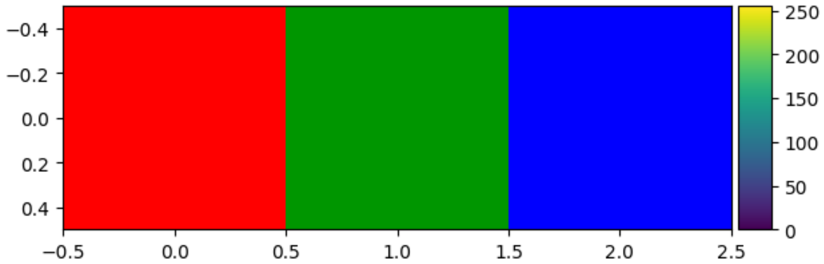
Remember that you do not have to manually encode each number!

The below image was constructed with the use of the arange function of NumPy and created another image by getting the transpose of the first one.

```
array_spectrum = np.array([np.arange(0,255,15),
                           np.arange(255,0,-15),
                           np.arange(0,255,15),
                           np.arange(255,0,-15)])
fig, ax = plt.subplots(1, 2, figsize=(12,4))
ax[0].imshow(array_spectrum, cmap = 'gray')
ax[0].set_title('Arange Generation')
ax[1].imshow(array_spectrum.T, cmap = 'gray')
ax[1].set_title('Transpose Generation');
```



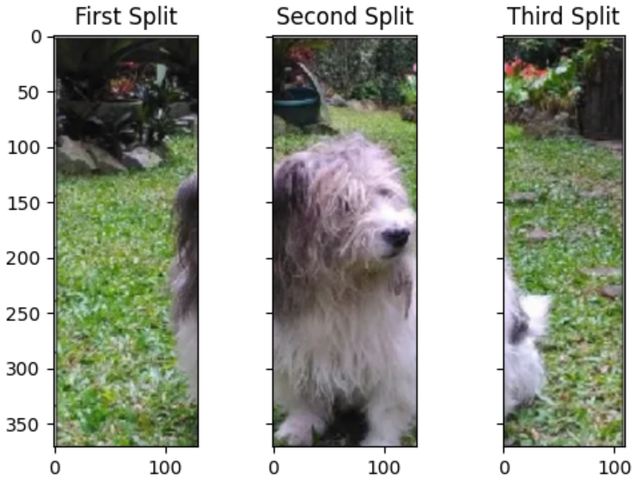
```
# Color Matrix
array_colors = np.array([[[255, 0, 0],
                          [0, 255, 0],
                          [0, 0, 255]]])
imshow(array_colors);
```



```
doggo = imread('doggo.PNG')
imshow(doggo);
doggo.shape # height, width, channels
```

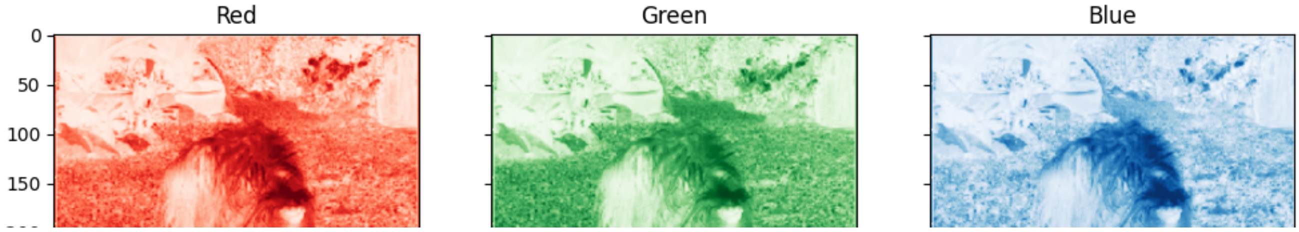
```
(371, 369, 4)
```

```
fig, ax = plt.subplots(1, 3,
                       figsize=(6,4),
                       sharey= True)

ax[0].imshow(doggo[:, 0:130])
ax[0].set_title('First Split')

ax[1].imshow(doggo[:, 130:260])
ax[1].set_title('Second Split')

ax[2].imshow(doggo[:, 260:390])
ax[2].set_title('Third Split');
```



```
imshow(doggo[95:250, 130:275]);
```

```
fig, ax = plt.subplots(1, 3, figsize=(12,4), sharey = True)
ax[0].imshow(doggo[:,:,0], cmap='Reds')
ax[0].set_title('Red')
ax[1].imshow(doggo[:,:,1], cmap='Greens')
ax[1].set_title('Green')
ax[2].imshow(doggo[:,:,2], cmap='Blues')
ax[2].set_title('Blue');
```



```
doggo_gray = rgb2gray(doggo)
fig, ax = plt.subplots(1, 5, figsize=(17,6), sharey = True)
ax[0].imshow(doggo_gray, cmap = 'gray')
ax[0].set_title('Grayscale Original')
ax[1].imshow(img_as_uint(doggo_gray > 0.25),
             cmap = 'gray')
ax[1].set_title('Greater than 0.25')
ax[2].imshow(img_as_uint(doggo_gray > 0.50),
             cmap = 'gray')
ax[2].set_title('Greater than 0.50');
ax[3].imshow(img_as_uint(doggo_gray > 0.75),
             cmap = 'gray')
ax[3].set_title('Greater than 0.75');
ax[4].imshow(img_as_uint(doggo_gray > np.mean(doggo_gray)),
             cmap = 'gray')
ax[4].set_title('Greater than Mean');
```