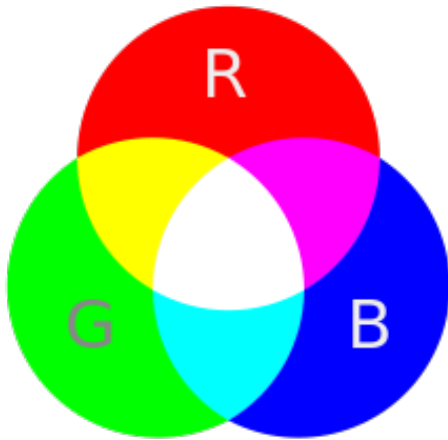


# PIXELS



1 pixel is a 3 element list  
the first element is the value for the Red color channel (i.e. the amt of Red that pixel has. This value is in the range  $[0.0, 1.0]$ ).

The second element is the green color channel value, and the third element is the blue color channel value. All together, we represent a pixel like this:

$$\text{pixel} = [0.0, 0.75, 1.0]$$

We can access a specific color channel using its appropriate index in the list.  
[Red, green, blue]  
0      1      2

# IMAGES

Images are made up of one or more pixels.

1x1 image (smallest possible)



2x2 image: 

RGB	RGB
RGB	RGB

 (4 pixels)  
2 rows, 2 cols

1x3 image: 

RGB	RGB	RGB
-----	-----	-----

 (3 pixels)  
1 row, 3 cols

The size of an image is the amount of pixels the image has.

We denote the size by writing the height by the width (h x w). The height is the amount of rows an image has. The width is the amount of columns.

3x4 image: 

	0	1	2	3
0	RGB	RGB	RGB	RGB
1	RGB	RGB	RGB	RGB
2	RGB	RGB	RGB	RGB

 12 pixels

Therefore, in order to represent images in Python, we need to use a 3D array.

$$\text{image} = \left[ \left[ \left[ 0.0, 1.0, 0.5 \right], \left[ 1.0, 0.0, 0.5 \right] \right], \left[ \left[ 1.0, 1.0, 1.0 \right], \left[ 1.0, 0.0, 1.0 \right] \right] \right]$$

=

0.0, 1.0, 0.5	1.0, 0.0, 0.5
1.0, 1.0, 1.0	1.0, 0.0, 1.0

BUT we don't want to define each pixel explicitly like above. Instead, we can use

`np.ones` or `np.zeros` to create ALL WHITE or ALL BLACK images.

`np.ones((3, 2, 3))`

1.0, 1.0, 1.0	1.0, 1.0, 1.0
1.0, 1.0, 1.0	1.0, 1.0, 1.0
1.0, 1.0, 1.0	1.0, 1.0, 1.0

each pixel = `[1.0, 1.0, 1.0]`

`np.zeros((1, 4, 3))`

0.0, 0.0, 0.0	0.0, 0.0, 0.0	0.0, 0.0, 0.0	0.0, 0.0, 0.0
---------------	---------------	---------------	---------------

each pixel = `[0.0, 0.0, 0.0]`