In [2]:
```
import cv2
import numpy as np
```

Grayscaling is a process by which an image is converted from a full color to shades of gray (black & white)

In OpenCV, many functions grayscale images before processing, This done because it simplifies the image, acting almost as noise reduction and increasing processing time as there is less information in the image.

Let convert our color image to grayscale

Method 1:-

cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

it is used to convert the image in gray scale format

In [1]:
```
#Load our input image
import numpy as np, cv2
img = cv2.imread('my.jpeg')
cv2.imshow('Original', img)

#We use cvtColor, to convert to gray scale
gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imshow('Grayscaled Image', gray_image)
cv2.waitKey()
cv2.destroyAllWindows()
```

In [2]:
```
print(f'Shape of colored image {img.shape}')
print(f'Shape of gray image {gray_image.shape}')
```

```
Shape of colored image (5184, 3456, 3)
Shape of gray image (5184, 3456)
```

Method 2:- it is faster than 1st method in typing not in processing because it's code length is very less than the first one

In [3]:
```
img = cv2.imread('my.jpeg', 0) #By changing the number inside height and width also change
cv2.imshow('Grayscale', img)
cv2.waitKey()
cv2.destroyAllWindows()
```

In [4]:
```
print(f'Shape of gray image image {img.shape}')
```

```
Shape of gray image image (5184, 3456)
```

In [5]:
```
img = cv2.imread('my.jpeg', 200000) #By changing the number inside height and width also change
cv2.imshow('Grayscale', img)
cv2.waitKey()
cv2.destroyAllWindows()
```

In [6]:
```
print(f'Shape of gray image image {img.shape}')
```

```
Shape of gray image image (648, 432)
```

write image

In [14]:
```
cv2.imwrite('grayimage.jpeg', img)
```

Out[14]: True

In [17]:
```
img = cv2.imread('grayimage.jpeg')
cv2.imshow('GrayImage',img)
cv2.waitKey()
cv2.destroyAllWindows()
```