Library Open CV

```
import numpy as np
import cv2
from google.colab.patches import cv2_imshow

#How to read an image
img = cv2.imread("/content/fruits.png")
print(type(img))#foe open cv it is numpy array
print(img.shape)
# 785 * 1152 * 3
# 785 and 1152 is image resolution and 3 is the channel
print(img)
```

```
<class 'NoneType'>
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-5-cf5d491d28ac> in <module>
      2 img = cv2.imread("/content/fruits.png")
      3 print(type(img))#foe open cv it is numpy array
----> 4 print(img.shape)
      5 # 785 * 1152 * 3
      6 # 785 and 1152 is image resolution and 3 is the channel

AttributeError: 'NoneType' object has no attribute 'shape'
```

SEARCH STACK OVERFLOW

```
cv2_imshow(img)
cv2.waitKey(0)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-6-e9648498cb8e> in <module>
----> 1 cv2_imshow(img)
      2 cv2.waitKey(0)

/usr/local/lib/python3.9/dist-packages/google/colab/patches/__init__.py in
cv2_imshow(a)
     16       (N, M, 4) is an NxM BGRA color image.
     17   """
---> 18   a = a.clip(0, 255).astype('uint8')
     19   # cv2 stores colors as BGR; convert to RGB
     20   if a.ndim == 3:

AttributeError: 'NoneType' object has no attribute 'clip'
```

SEARCH STACK OVERFLOW

CONVERTING A RGB IMAGE INTO A GRAYSCALE IMAGE

```
img_gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
img_gray
cv2_imshow(img_gray)
cv2.waitKey(0)
```

```
---------------------------------------------------------------------------
error                                     Traceback (most recent call last)
<ipython-input-7-812773ece646> in <module>
```

```
print(img_gray.shape)
```

```
(785, 1152)
error: OpenCV(4.6.0) /io/opencv/modules/imgproc/src/color.cpp:182: error: (-215:Assertion failed) !_src.empty() in function
```

```
print(img)
```

```
[[[0 0 0]
  [0 0 0]
  [0 0 0]
  ...
  [0 0 0]
  [0 0 0]
  [0 0 0]]

 [[0 0 0]
  [0 0 0]
  [0 0 0]
  ...
  [0 0 0]
  [0 0 0]
  [0 0 0]]

 [[0 0 0]
  [0 0 0]
  [0 0 0]
  ...
  [0 0 0]
  [0 0 0]
  [0 0 0]]

 ...

 [[0 0 0]
  [0 0 0]
  [0 0 0]
  ...
  [0 0 0]
  [0 0 0]
  [0 0 0]]

 [[0 0 0]
  [0 0 0]
  [0 0 0]
  ...
  [0 0 0]
  [0 0 0]
  [0 0 0]]

 [[0 0 0]
  [0 0 0]
  [0 0 0]
  ...
  [0 0 0]
  [0 0 0]
  [0 0 0]]]
```

```
imgBlue = img[:,:,0]
imgGreen = img [:,:,1]
imgRed = img[:,:,2]

new_image = np.hstack((imgBlue,imgGreen,imgRed))

cv2_imshow(new_image)
cv2.waitKey(0)
```

RESIZE

```
img_resize = cv2.resize(img,(256,256))
```

```
cv2_imshow(img_resize)
cv2.waitKey(0)
```

```
#when you dont know the size of  the  image
image_resize1 = cv2.resize(img,(img.shape[1]//2,img.shape[0]//2))
cv2_imshow(image_resize1)
cv2.waitKey(0)
```

Image Flip

```
#1 means horizontal Flip
img_flip = cv2.flip(img,1)
cv2_imshow(img_flip)
cv2.waitKey(0)
```

-1

```
#0 means vertical Flip
img_flip1 = cv2.flip(img,0)
cv2_imshow(img_flip1)
cv2.waitKey(0)
```

```
#-1 means horizontal and vertical Flip
img_flip2 = cv2.flip(img,-1)
cv2_imshow(img_flip2)
cv2.waitKey(0)
```

```
-1
```

CROPPING

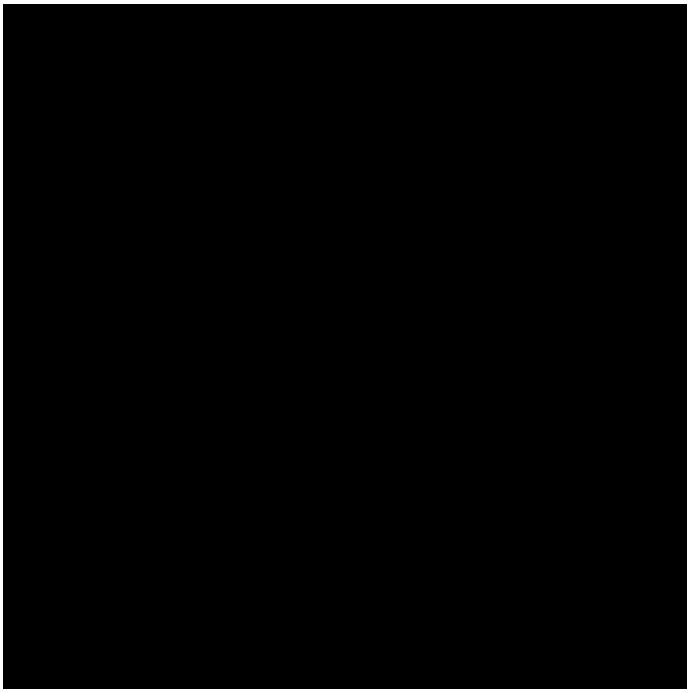
```
img_crop = img[100:300,200:500]
#first height and second is width
cv2_imshow(img_crop)
cv2.waitKey(0)
```

```
#saving the reduced size image
#Saving using imwrite
cv2.imwrite('fruit_image_reducedsize.png',img_crop)
```

```
True
```

DRAWING SHAPES AND TEXT ON IMAGES

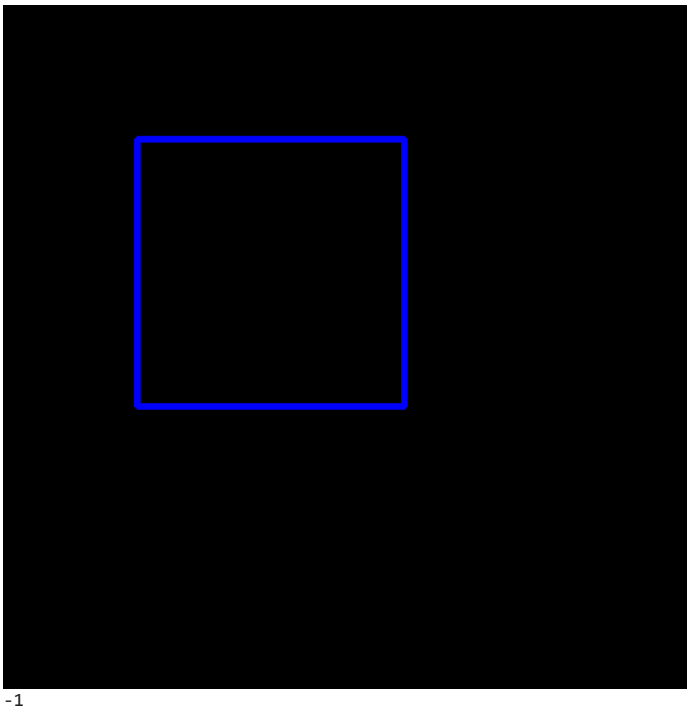
```
img = np.zeros((512,512,3))
cv2_imshow(img)
cv2.waitKey(0)
```



```
-1
```

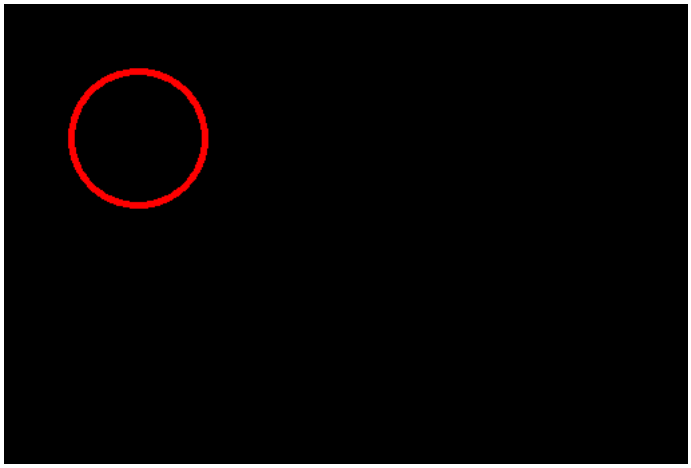
```
#Rectangle
#we are constructing the rectangle where
#(image, point 1 in pixels, point2 in pixels, color(B,G,R(255,0,0)))u
cv2.rectangle(img, pt1 = (100,100), pt2= (300,300), color = (255,0,0
# thickness = -1 means entire square rectangle will be covered with t
# thickness = 0 means only border will be there
cv2_imshow(img)
cv2.waitKey(0)
```

```
#Rectangle
#we are constructing the rectangle where
#(image, point 1 in pixels, point2 in pixels, color(B,G,R(255,0,0)))
img3 = np.zeros((512,512,3))
cv2.rectangle(img3, pt1 = (100,100), pt2= (300,300), color = (255,0,
# thickness = 3  positive numbers means only borders will be covered
cv2_imshow(img3)
cv2.waitKey(0)
```

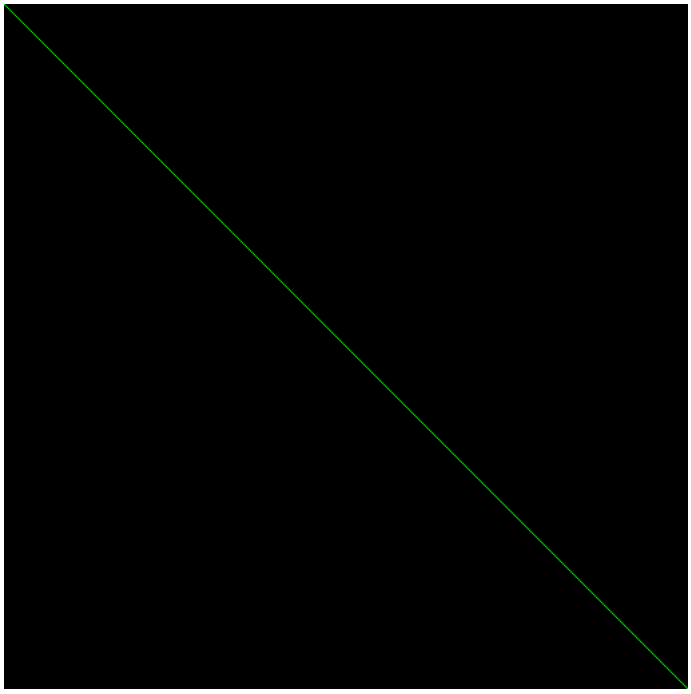


-1

```
# Drawing a circle
img4 = np.zeros((512,512,3))
#
cv2.circle(img4, center = (100,100), radius = 50 , color = (0,0,255)
cv2_imshow(img4)
cv2.waitKey(0)
# wait key = 0 means we do it infinity
```



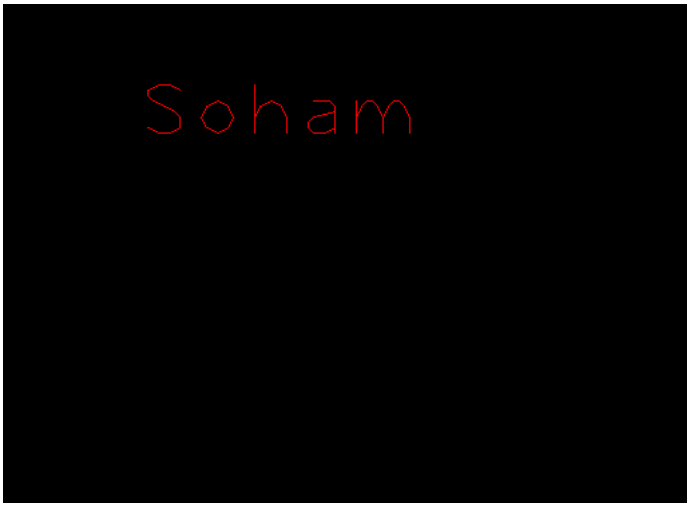
```
# Drawing a Line
img5 = np.zeros((512,512,3))
#
cv2.line(img5, pt1 = (0,0), pt2 = (512,512), color = (0,255,0)  )
cv2_imshow(img5)
cv2.waitKey(0)
# wait key = 0 means we do it infinity
```



```
-1
```

TEXT AND FONT CREATION

```
img6 = np.zeros((512,512,3))
cv2.putText(img6, org =(100,100), text = "Soham", fontScale = 4, fon
cv2_imshow(img6)
cv2.waitKey(0)
```

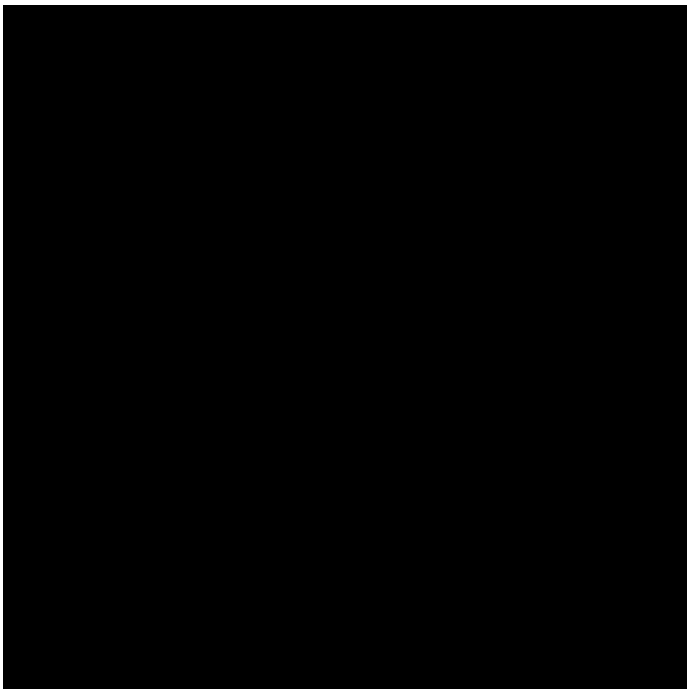


#WORKING WITH OPENCV EVENTS

INTERACTING WITH IMAGE WITH THE MOUSE CLICK OR SOME OTHER THING

```
img7 = np.zeros((512,512,3))
cv2_imshow(img7)
cv2.waitKey(0)

while TRUE :
  cv2_imshow(img7)
  if cv2.waitKey(1)
```



```
-1
```