import numpy as np  
import cv2 as cv

# Create lines:

# Crea una imagen en negro  
img = np.zeros((512,512,3), np.uint8)  
# Dibuja una línea azul diagonal con un grosor de 5 px  
cv.line (img,(0,0),(511,511),(255,0,0),5)  
  
# Muestra la imagen dibujada  
cv.imshow('Imagen Dibujada', img)  
k= cv.waitKey(0)  
if k == ord("s"):  
 cv.destroyAllWindows()

# Create rectangle

import numpy as np  
import cv2 as cv  
img = np.zeros((512,512,3), np.uint8)  
#(image where draw, corner up-left,corner down-right,color (RGB), pixel size of trazo)   
cv.rectangle (img,(384,0),(510,128),(0,255,0),3)  
# Muestra la imagen dibujada  
cv.imshow('Imagen Dibujada', img)  
k= cv.waitKey(0)  
if k == ord("s"):  
 cv.destroyAllWindows()

qt.qpa.plugin: Could not find the Qt platform plugin "wayland" in "/home/dread/miniconda3/envs/tf/lib/python3.9/site-packages/cv2/qt/plugins"

# Create Circle

import numpy as np  
import cv2 as cv  
img = np.zeros((512,512,3), np.uint8)  
#(image where draw,coordenade of center of circle,radius,color (RGB), pixel size of trazo)   
# if last number is negatve is filled  
cv.circle (img,(447,63), 63, (0,0,255), -1)  
# Muestra la imagen dibujada  
cv.imshow('Imagen Dibujada', img)  
k= cv.waitKey(0)  
if k == ord("s"):  
 cv.destroyAllWindows()

# Create Elipse

import numpy as np  
import cv2 as cv  
#(image where draw,central ubication of elipse(x,y),longitude de los ejes (longitud del eje mayor, longitud del eje menor). El ángulo de rotación de la elipse en sentido antihorario(startAngle , endAngle) denotan el inicio y el final del arco de elipse.   
img = np.zeros((512,512,3), np.uint8)  
cv.ellipse(img,(256,256),(100,50),0,0,180,255,-1)  
# Muestra la imagen dibujada  
cv.imshow('Imagen Dibujada', img)  
k= cv.waitKey(0)  
if k == ord("s"):  
 cv.destroyAllWindows()

# Polygon

import numpy as np  
import cv2 as cv  
img = np.zeros((512,512,3), np.uint8)  
  
pts = np.array([[10,5],[20,30],[70,20],[50,10]], np.int32)  
pts = pts.reshape((-1,1,2))  
cv.polylines (img,[pts], True ,(0,255,255))  
# Muestra la imagen dibujada  
cv.imshow('Imagen Dibujada', img)  
k= cv.waitKey(0)  
if k == ord("s"):  
 cv.destroyAllWindows()

# Add Text

import numpy as np  
import cv2 as cv  
img = np.zeros((512,512,3), np.uint8)  
# (image where write,text,begin to write(x,y),font,scale,color,2,)  
font = cv.FONT\_HERSHEY\_SIMPLEX  
cv.putText(img,'OpenCV',(10,500), font, 4,(255,255,255),2,cv.LINE\_AA)  
cv.imshow('Imagen Dibujada', img)  
k= cv.waitKey(0)  
if k == ord("s"):  
 cv.destroyAllWindows()

Ejercicio

import cv2  
import numpy as np  
  
image =np.full((512,512,3),255,dtype=np.uint8)  
cv2.ellipse(image,(256,200),(90,90),120,0,300,(0,0,255),-1)  
cv2.ellipse(image,(256,200),(30,30),120,0,300,(255,255,255),-1)  
  
cv2.ellipse(image,(160,400),(90,90),0,0,300,(0,255,0),-1)  
cv2.ellipse(image,(160,400),(30,30),0,0,300,(255,255,255),-1)  
  
cv2.ellipse(image,(360,400),(90,90),300,0,300,(255,0,0),-1)  
cv2.ellipse(image,(360,400),(30,30),300,0,300,(255,255,255),-1)  
  
cv2.putText(image,'OpenCV',(200,70),cv2.FONT\_HERSHEY\_SIMPLEX,1.5,(0,0,0),6)  
  
cv2.imwrite("resources/opencv\_logoGenerate.png",image)  
cv2.imshow("image",image)  
cv2.waitKey(0)

-1