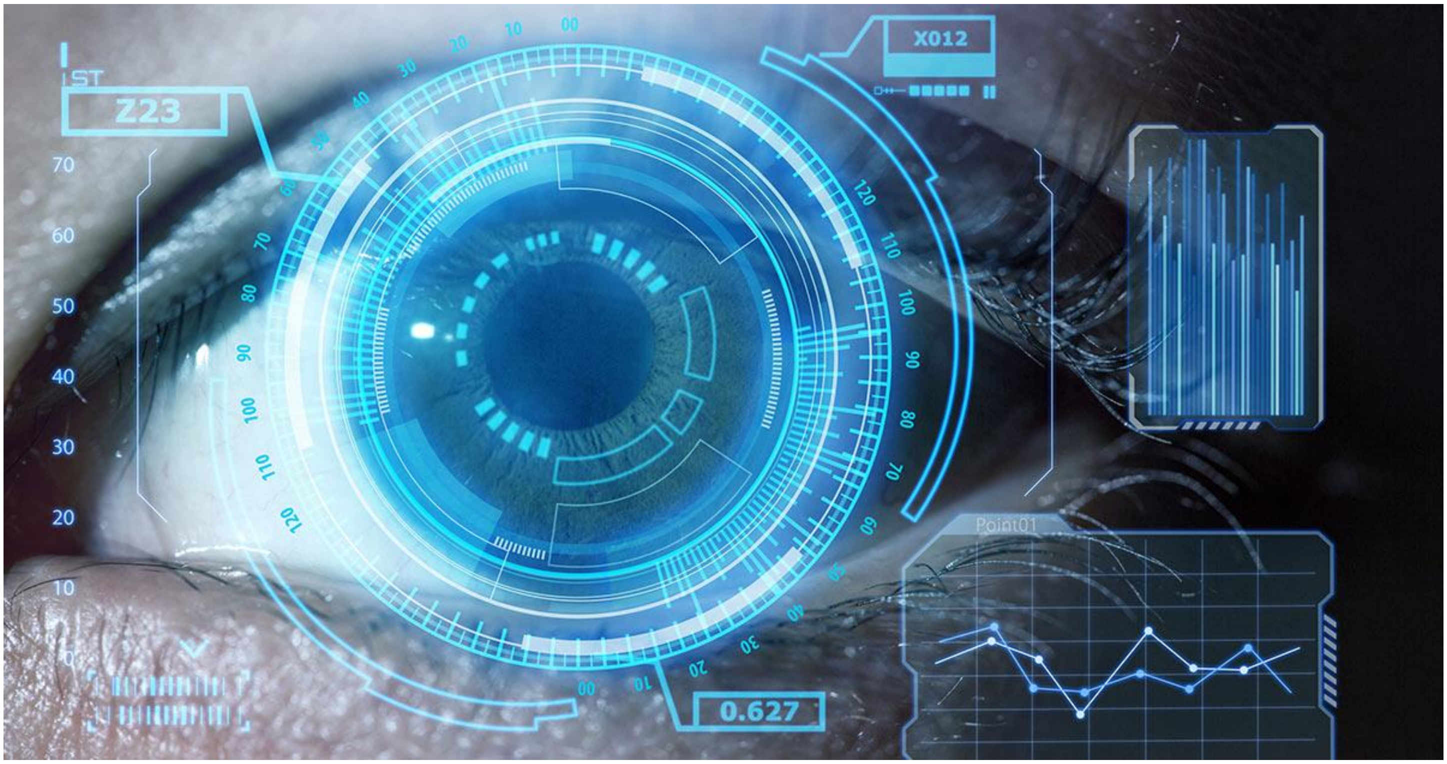
“COMPUTER VISION”

ENG: AHMED MUBARAK

01020451375



SESSION NO.“6”

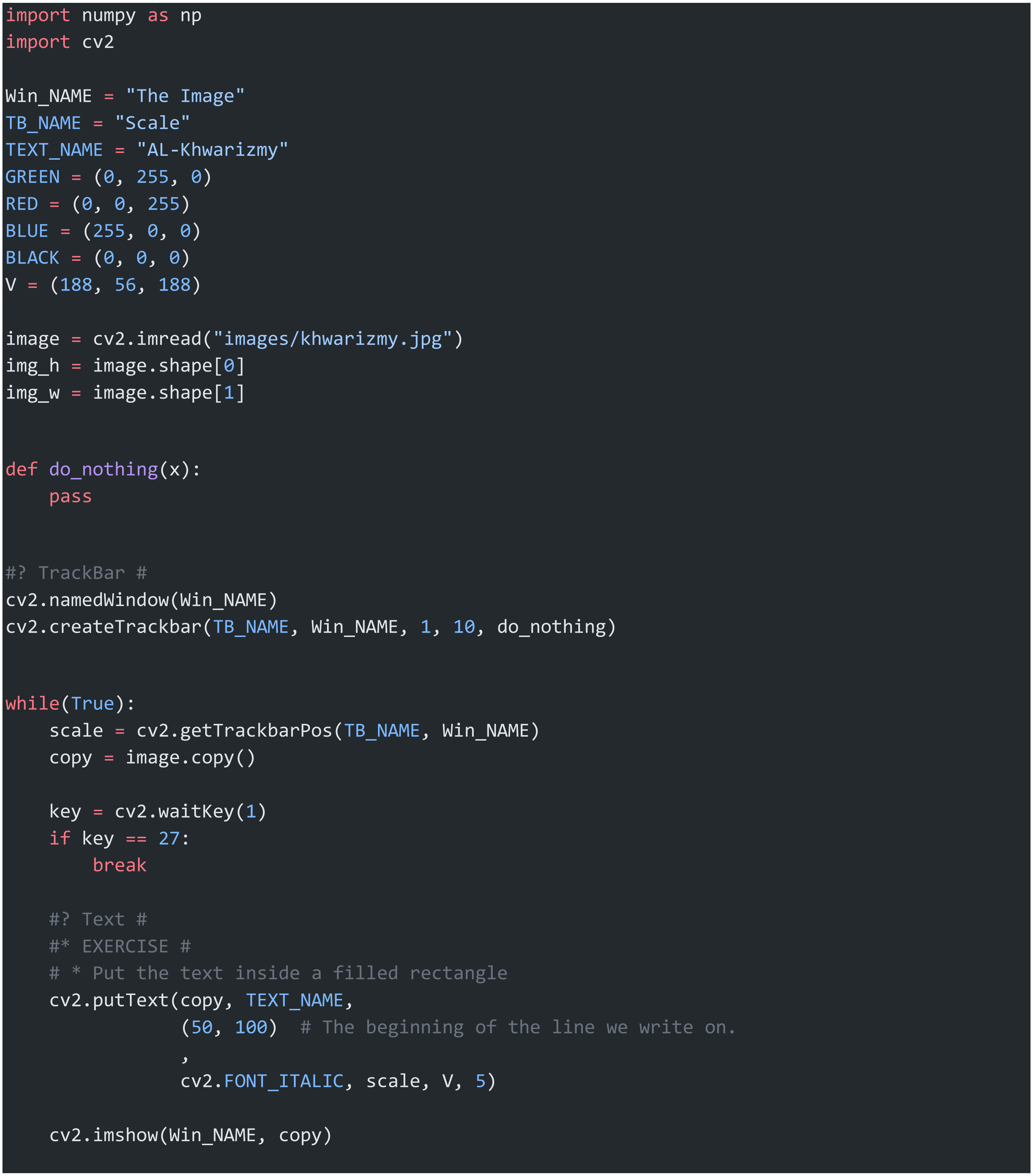
- OPENCV LIBRARY

  1. Text & Trackbar
  2. Face recognition using text & trackbar
  3. Fonts
  4. WaitKey

ENG.AHMED MUBARAK

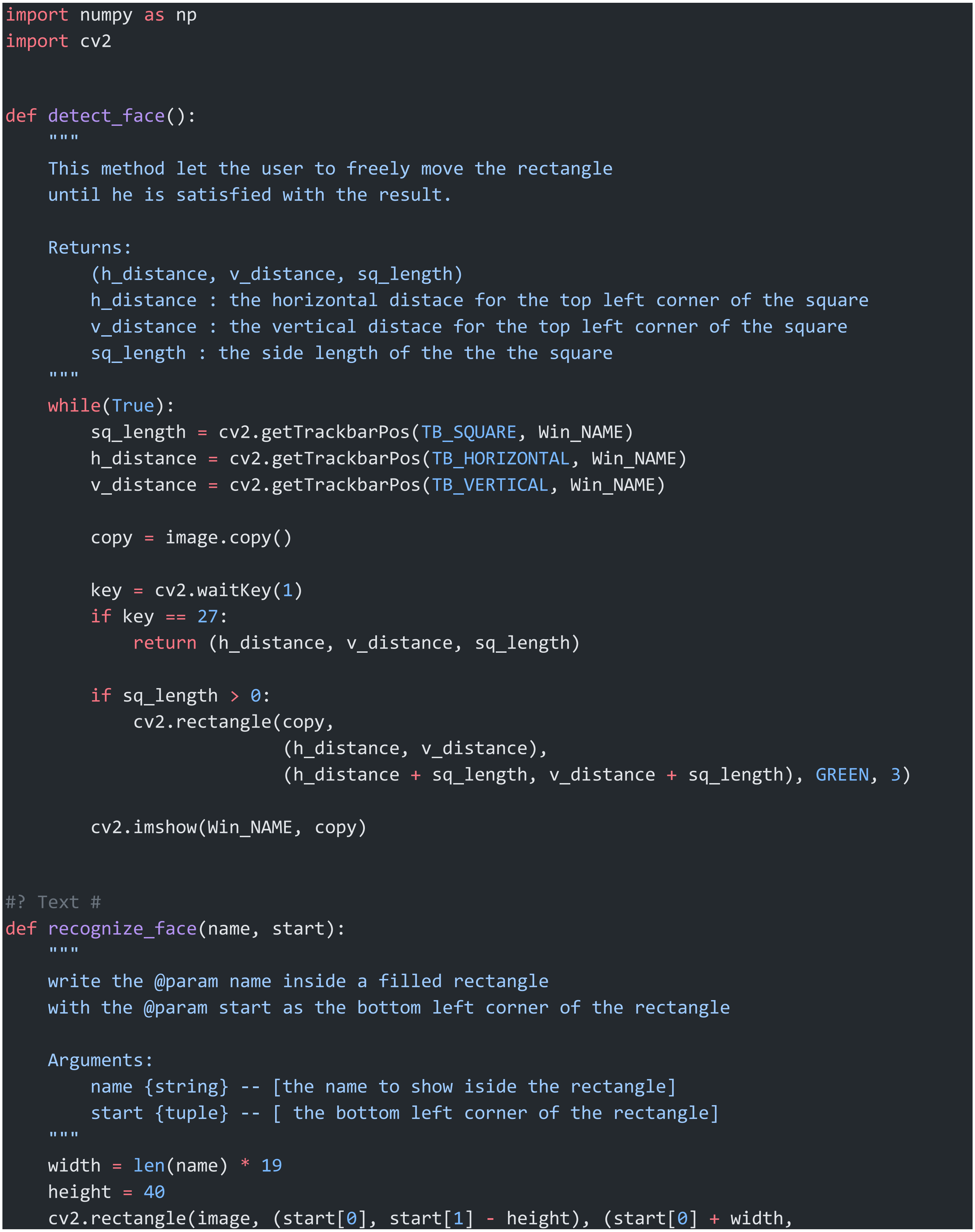
01020451375

1. Text & Trackbar



```
import numpy as np
import cv2

Win_NAME = "The Image"
TB_NAME = "Scale"
TEXT_NAME = "AL-Khwarizmy"
GREEN = (0, 255, 0)
RED = (0, 0, 255)
BLUE = (255, 0, 0)
BLACK = (0, 0, 0)
V = (188, 56, 188)

image = cv2.imread("images/khwarizmy.jpg")
img_h = image.shape[0]
img_w = image.shape[1]


def do_nothing(x):
    pass


#? TrackBar #
cv2.namedWindow(Win_NAME)
cv2.createTrackbar(TB_NAME, Win_NAME, 1, 10, do_nothing)


while(True):
    scale = cv2.getTrackbarPos(TB_NAME, Win_NAME)
    copy = image.copy()

    key = cv2.waitKey(1)
    if key == 27:
        break

    #? Text #
    #* EXERCISE #
    # * Put the text inside a filled rectangle
    cv2.putText(copy, TEXT_NAME,
                (50, 100)  # The beginning of the line we write on.
                ,
                cv2.FONT_ITALIC, scale, V, 5)

    cv2.imshow(Win_NAME, copy)
```

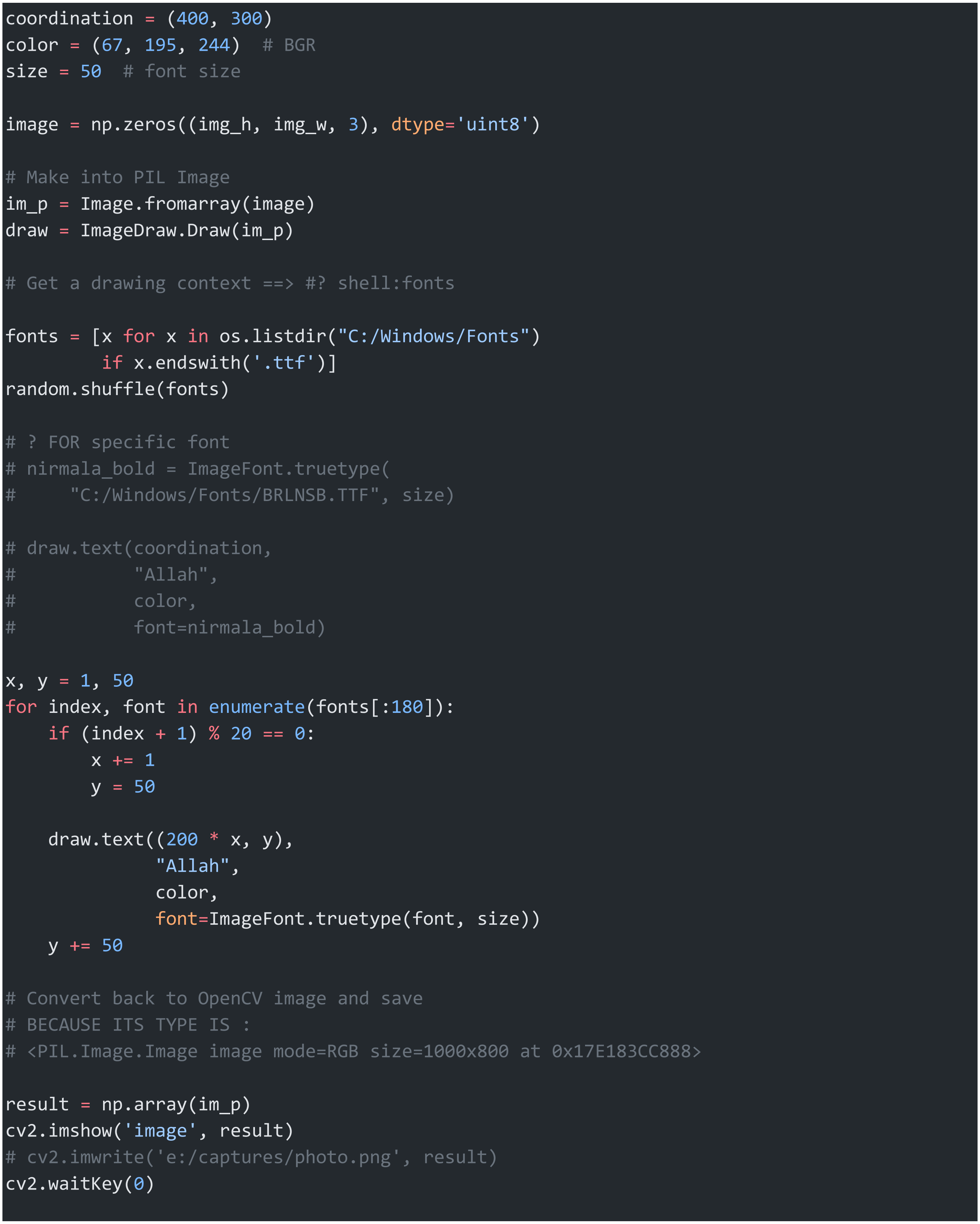
2. Face recognition using text & trackbar



```
import numpy as np
import cv2


def detect_face():
    """
    This method let the user to freely move the rectangle
    until he is satisfied with the result.

    Returns:
        (h_distance, v_distance, sq_length)
        h_distance : the horizontal distace for the top left corner of the square
        v_distance : the vertical distace for the top left corner of the square
        sq_length : the side length of the the the square
    """
    while(True):
        sq_length = cv2.getTrackbarPos(TB_SQUARE, Win_NAME)
        h_distance = cv2.getTrackbarPos(TB_HORIZONTAL, Win_NAME)
        v_distance = cv2.getTrackbarPos(TB_VERTICAL, Win_NAME)

        copy = image.copy()

        key = cv2.waitKey(1)
        if key == 27:
            return (h_distance, v_distance, sq_length)

        if sq_length > 0:
            cv2.rectangle(copy,
                          (h_distance, v_distance),
                          (h_distance + sq_length, v_distance + sq_length), GREEN, 3)

        cv2.imshow(Win_NAME, copy)


#? Text #
def recognize_face(name, start):
    """
    write the @param name inside a filled rectangle
    with the @param start as the bottom left corner of the rectangle

    Arguments:
        name {string} -- [the name to show iside the rectangle]
        start {tuple} -- [ the bottom left corner of the rectangle]
    """
    width = len(name) * 19
    height = 40
    cv2.rectangle(image, (start[0], start[1] - height), (start[0] + width,
                                                         start[1]), GREEN, -1)
    cv2.putText(image, name, (start[0], start[1] - 10),
                cv2.FONT_ITALIC, 0.75, BLACK, 1)


Win_NAME = "The Image"
TB_SQUARE = "Square Length"
TB_HORIZONTAL = "Horizontal Distance"
TB_VERTICAL = "Vertical Distance"
GREEN = (0, 255, 0)
BLACK = (0, 0, 0)

image = cv2.imread("images/khwarizmy.jpg")
img_h = image.shape[0]
img_w = image.shape[1]


#? TrackBar #
cv2.namedWindow(Win_NAME)
cv2.createTrackbar(TB_SQUARE, Win_NAME, 100, 400, lambda x: x)
cv2.createTrackbar(TB_HORIZONTAL, Win_NAME, 0, img_w, lambda x: x)
cv2.createTrackbar(TB_VERTICAL, Win_NAME, 0, img_h, lambda x: x)


h_distance, v_distance, sq_length = detect_face()
cv2.rectangle(image,
              (h_distance, v_distance),
              (h_distance + sq_length, v_distance + sq_length), GREEN, 3)
recognize_face("Al-Khwarizmy", (h_distance, v_distance))
cv2.imshow(Win_NAME, image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

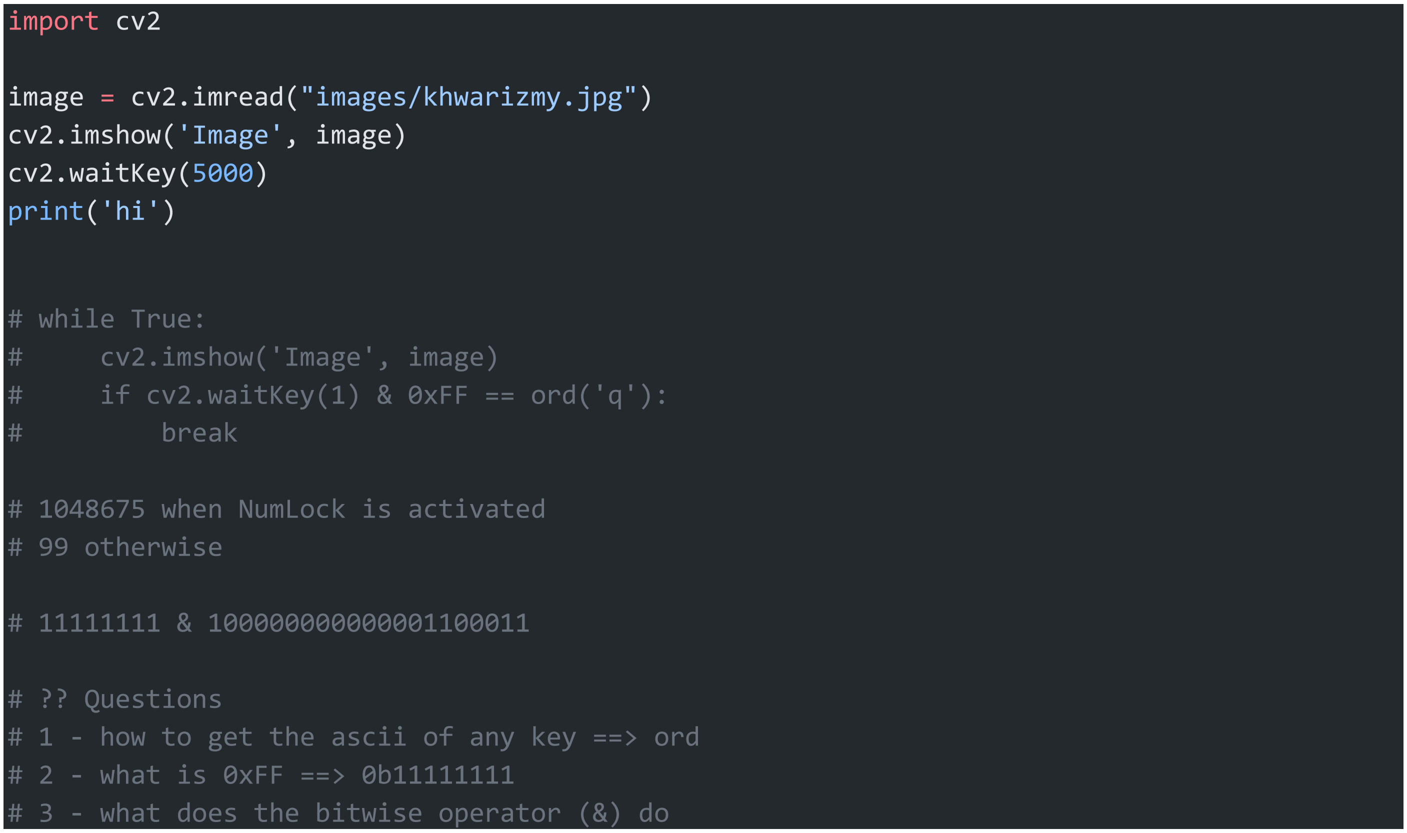
3. Fonts

```
# check first if the PIL is installed
#! press (ctrl + z) to exit python

# ? python -m pip install Pillow
# ? conda install -c anaconda pillow

from PIL import Image, ImageFont, ImageDraw
import numpy as np
import random
import cv2
import os

img_w = 1920
img_h = 1080
```



```python
coordination = (400, 300)
color = (67, 195, 244)  # BGR
size = 50  # font size

image = np.zeros((img_h, img_w, 3), dtype='uint8')

# Make into PIL Image
im_p = Image.fromarray(image)
draw = ImageDraw.Draw(im_p)

# Get a drawing context ==> #? shell:fonts

fonts = [x for x in os.listdir("C:/Windows/Fonts")
         if x.endswith('.ttf')]
random.shuffle(fonts)

# ? FOR specific font
# nirmala_bold = ImageFont.truetype(
#     "C:/Windows/Fonts/BRLNSB.TTF", size)

# draw.text(coordination,
#            "Allah",
#            color,
#            font=nirmala_bold)

x, y = 1, 50
for index, font in enumerate(fonts[:180]):
    if (index + 1) % 20 == 0:
        x += 1
        y = 50

    draw.text((200 * x, y),
              "Allah",
              color,
              font=ImageFont.truetype(font, size))
    y += 50

# Convert back to OpenCV image and save
# BECAUSE ITS TYPE IS :
# <PIL.Image.Image image mode=RGB size=1000x800 at 0x17E183CC888>

result = np.array(im_p)
cv2.imshow('image', result)
# cv2.imwrite('e:/captures/photo.png', result)
cv2.waitKey(0)
```

4. waitKey



```
import cv2

image = cv2.imread("images/khwarizmy.jpg")
cv2.imshow('Image', image)
cv2.waitKey(5000)
print('hi')


# while True:
#     cv2.imshow('Image', image)
#     if cv2.waitKey(1) & 0xFF == ord('q'):
#         break

# 1048675 when NumLock is activated
# 99 otherwise

# 11111111 & 100000000000001100011

# ?? Questions
# 1 - how to get the ascii of any key ==> ord
# 2 - what is 0xFF ==> 0b11111111
# 3 - what does the bitwise operator (&) do
```

WITH MY BEST WISHES

ENG/AHMED MUBARAK 😉