**#careful with apostrophes, they will be wrong in Pycharm if I type them here, copy from others. Also indents cause problems: write code in Pycharm, copy it and simply delete the big linespaces**

**#Import**:  
import numpy as np  
import cv2

**#Read, show, write image**:  
img = cv2.imread('/home/philipp/Desktop/image.jpg', 1) #1=color, 0=gray, -1 = no change  
cv2.imshow('windowName',img)  
cv2.waitKey(0)  
cv2.destroyAllWindows()  
cv2.imwrite('/home/philipp/Desktop/image.jpg', img)

**#convert color and show:**gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY) **#don’t forget to save the image! (here: gray)**cv2.imshow('windowName', gray)  
cv2.waitKey(0)  
cv2.destroyAllWindows()  
 **#Read, show video**:  
cap = cv2.VideoCapture('/home/philipp/Desktop/video.mp4')

while(cap.isOpened()):  
 ret, frame = cap.read()  
 cv2.imshow('windowName', frame)  
 if cv2.waitKey(30) & 0xFF == ord('q'): #Change frame-speed; close when q is pressed  
 break  
  
cap.release()  
cv2.destroyAllWindows()

**#Read, show, write video**: you can also write in mp4, there’s an error message but it works

cap = cv2.VideoCapture('/home/philipp/Desktop/video.mp4')  
fourcc = cv2.VideoWriter\_fourcc(\*'XVID') # Define the codec and create VideoWriter object (fourcc)  
  
x=0  
while(cap.isOpened()):  
 ret, frame = cap.read()  
  
 if ret==True:  
 if x==0:  
 out = cv2.VideoWriter('/home/philipp/Desktop/video\_written.avi',fourcc, 20.0,(frame.shape[1],frame.shape[0])) #define: format, fps, and frame-size (pixels)  
 x=1  
  
 cv2.imshow('windowName', frame)  
 out.write(frame)  
  
 if cv2.waitKey(10) & 0xFF == ord('q'):  
 break  
  
cap.release()  
cv2.destroyAllWindows()

**#median-blur an image**img\_blurred= cv2.medianBlur(img,5) # kernel-size: any odd number

**#flip an image**frame=cv2.flip(img,0) #flip around x: 0; y: 1; both: -1

**# Create a black image**

img = np.zeros((512,512,3), np.uint8)

**#Change a pixel’s color value:**

img[100,100] = [255,255,255]

**#Draw geometric figures into img**   
**#Image, position, color, thickness (-1 = filled)**cv2.line(img,(0,0),(511,511),(255,0,0),5)  
cv2.rectangle(img,(384,0),(510,128),(0,255,0),3)  
cv2.circle(img,(447,63), 63, (0,0,255), -1)  
cv2.ellipse(img,(256,256),(100,50),0,0,180,255,-1)

**#Draw free polyeder**pts = np.array([[10,5],[20,30],[70,20],[50,10]], np.int32)  
pts = pts.reshape((-1,1,2))  
img = cv2.polylines(img,[pts],True,(0,255,255))

**#Add Text**font = cv2.FONT\_HERSHEY\_TRIPLEX  
cv2.putText(img,'myText',(10,500), font, 4,(255,255,255),2,cv2.LINE\_AA)

**#Cut part of an image:**roi = img[20:100,20:100] #roi=region of interest

**#Any key to exit or ‘s’ to save and exit**

k = cv2.waitKey(0)  
if k == ord('s'):  
 cv2.imwrite('/home/philipp/Desktop/image.png',img)  
cv2.destroyAllWindows()

**#Merge 2 images**rows,cols,channels= img2.shape #must be the smaller image!  
roi = img1[0:rows, 0:cols] #the bigger image is cut down img2’s dimensions  
img3 = cv2.addWeighted(img2,0.5,roi,0.5,-20) #las argument = static Offset

**#Measure time**e1 = cv2.getTickCount()  
e2 = cv2.getTickCount()  
time = (e2 - e1)/ cv2.getTickFrequency()

**#Apply thresholding**; numbers: threshold, color to be displayed  
ret, img\_thresholded = cv2.threshold(img,150,255, cv2.THRESH\_BINARY) #important to put ret!

**#Background Subtraction**fgbg = cv2.createBackgroundSubtractorMOG2()  
fgmask = fgbg.apply(frame) #within the while-loop

**#Median-filtering**fgmask\_median= cv2.medianBlur(fgmask, 9) # number of neighbors

**#Show images in the same window 🡺stack them to one matrix**tmp1 = np.hstack((img1, img2))  
tmp2 = np.hstack((img3,img4))  
img = np.vstack((tmp1, tmp2))

**#Change Windowsize (window, not image!)**cv2.namedWindow('frame',cv2.WINDOW\_NORMAL)  
cv2.resizeWindow('frame', 800,500)  
cv2.imshow('frame',frame)

**#Extract certain colors**hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV) # define range of blue color in HSV  
lower\_blue = np.array([110,50,50])  
upper\_blue = np.array([130,255,255]) # Threshold the HSV image to get only blue colors  
mask = cv2.inRange(hsv, lower\_blue, upper\_blue)# Bitwise-AND mask and original image  
res = cv2.bitwise\_and(frame,frame, mask= mask)  
cv2.imshow('frame',frame)  
cv2.imshow('mask',mask)  
cv2.imshow('res',res)  
k = cv2.waitKey(5) & 0xFF  
if k == 27:  
break

**#Overwrite a previous output**import time  
for x in range (0,5):  
 b = "Loading" + "." \* x  
 print ("\r"+b,end= "")  
 time.sleep(1)

**#Plot 2 arrays vs. t**from matplotlib.pyplot import\*  
plot(t, N\_1, t, N\_2) #alternativ kann man auch nur plot(N\_1)  
show()

**#iterate over two indices**for i, j in np.ndindex(myMatrix.shape):

**#Create a matrix of a certain size**m = numpy.empty((3, 4))

**#Erosion/Dilation**kernel = np.ones((5,5),np.uint8)  
erosion = cv2.erode(frame,kernel,iterations = 1)  
dilation = cv2.dilate(frame,kernel,iterations = 1)