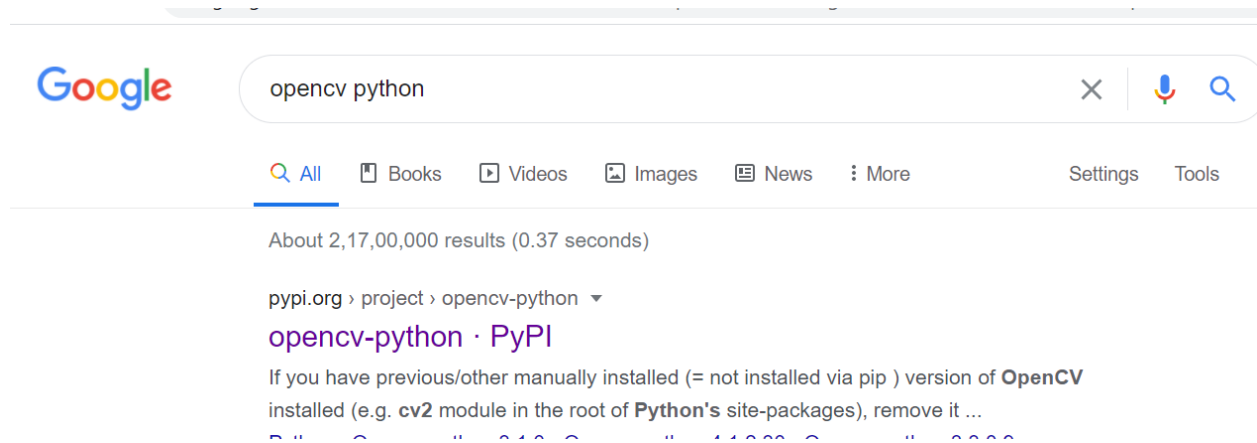
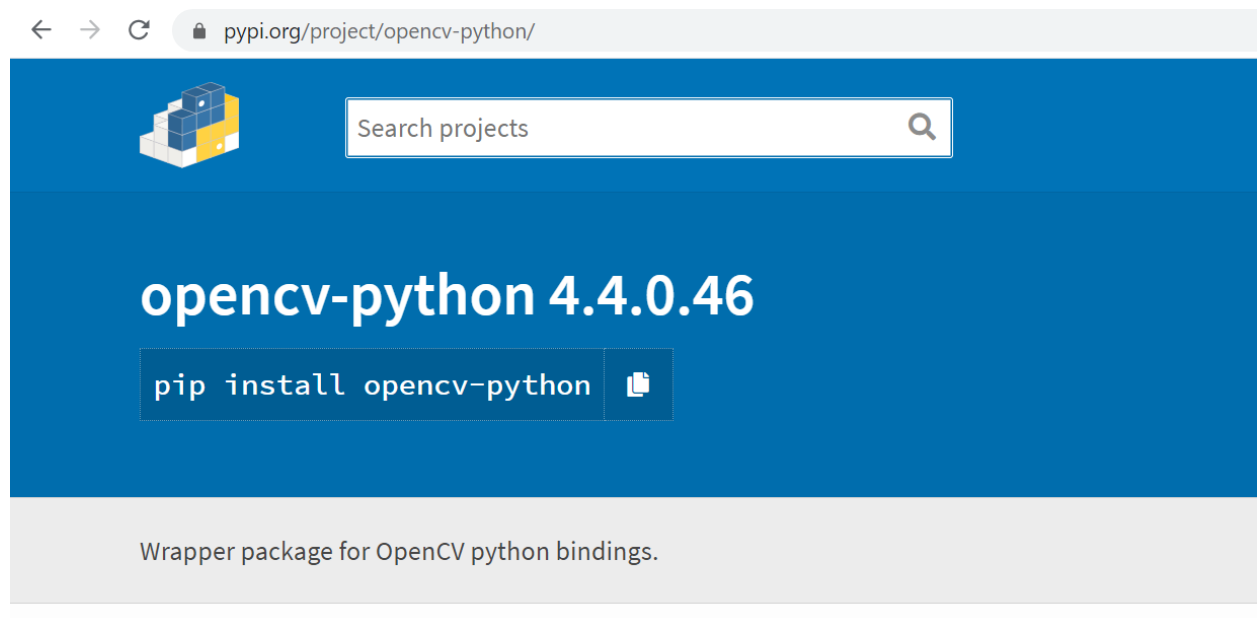


Step 1: Type opencv python in google



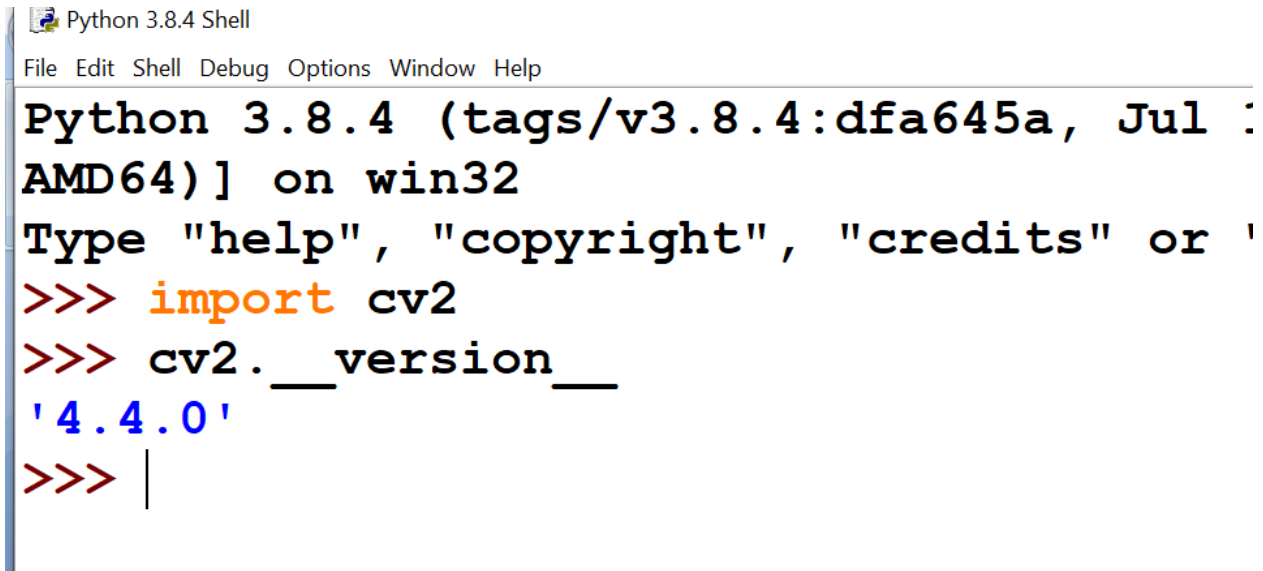
Step 2: Install below library in python



Step 3: Goto cmd and install

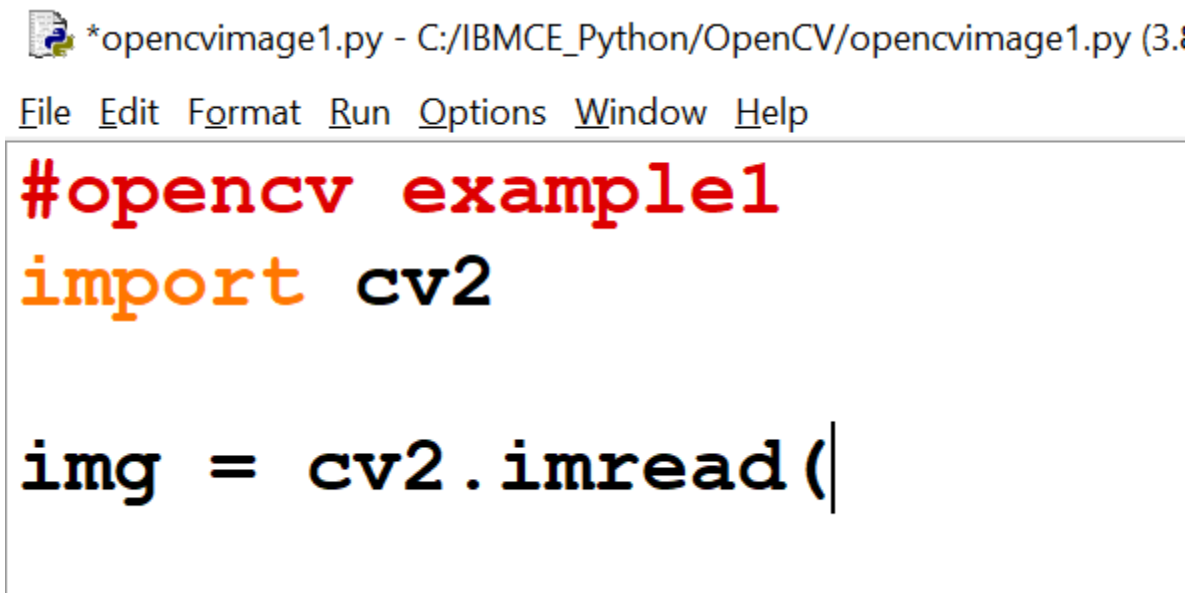


Step 4: import cv2 library and check for the version like below

A screenshot of a Python 3.8.4 Shell window. The title bar says "Python 3.8.4 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area shows the following code:

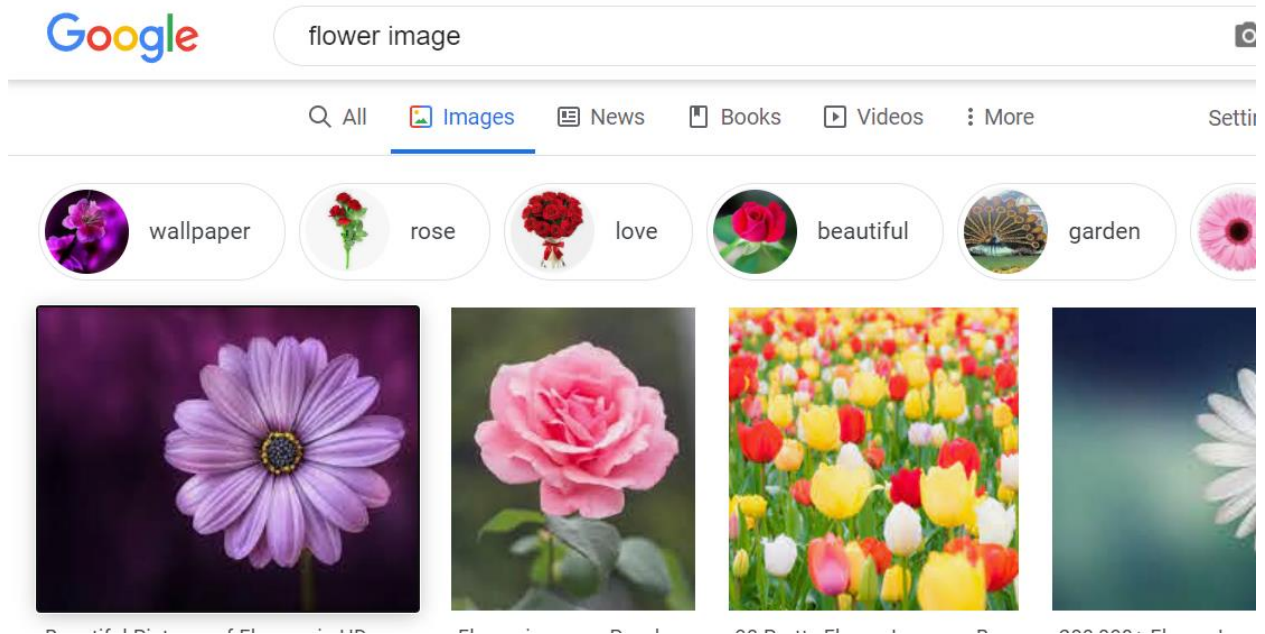
```
Python 3.8.4 (tags/v3.8.4:dfa645a, Jul :  
AMD64)] on win32  
Type "help", "copyright", "credits" or '  
>>> import cv2  
>>> cv2.__version__  
'4.4.0'  
>>> |
```

Step 5: Create a file in idle like below


A screenshot of a Python script editor window. The title bar says "*opencvimage1.py - C:/IBMCE_Python/OpenCV/opencvimage1.py (3.8.4)". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The main text area shows the following code:

```
#opencv example1  
import cv2  
  
img = cv2.imread(|
```

Step 6: Let us try to read an image, goto google and download a flower image as "flower1.jpg"



Step 7: Add below code and execute

 opencvimage1.py - C:/IBMCE_Python/OpenCV/opencvimage1.py (3.8.4)

File Edit Format Run Options Window Help

```
#opencv example1
```

```
import cv2
```


```
img = cv2.imread('flower1.jpg',0)
```

```
print(img)|
```

Step 8: When you execute you will see matrix like below

```
===== RESTART: C:/IBMCE_Python/OpenCV/opencvimage1.py =====
[[48 48 48 ... 18 18 18]
 [48 48 48 ... 19 19 19]
 [48 48 48 ... 20 20 20]
 ...
 [16 16 16 ... 12 12 12]
 [16 16 16 ... 12 12 12]
 [15 15 15 ... 12 12 12]]
\\>>> |
```

Step 9: Make flag as 1 and see output

 *opencvimage1.py - C:/IBMCE_Python/OpenCV/opencvimage1.py (3.8.4)*

File Edit Format Run Options Window Help

```
#opencv example1
import cv2

img = cv2.imread('flower1.jpg',1)

print(img)
```

Output is

```
>>>
===== RESTART: C:/IBMCE_Python/OpenCV/opencvimage1.py ==
[[[71 26 83]
  [71 26 83]
  [73 25 83]
  ...
  [30  8 32]
  [30  8 32]]
```

Step 10: include few more lines of code where we are display and deleting the image after 5 sec

```
#opencv example1
import cv2

#read the image in matrix form
img = cv2.imread('flower1.jpg',1)

cv2.imshow("flower1window",img)

cv2.waitKey(5000)
cv2.destroyAllWindows()

print(img)|
```

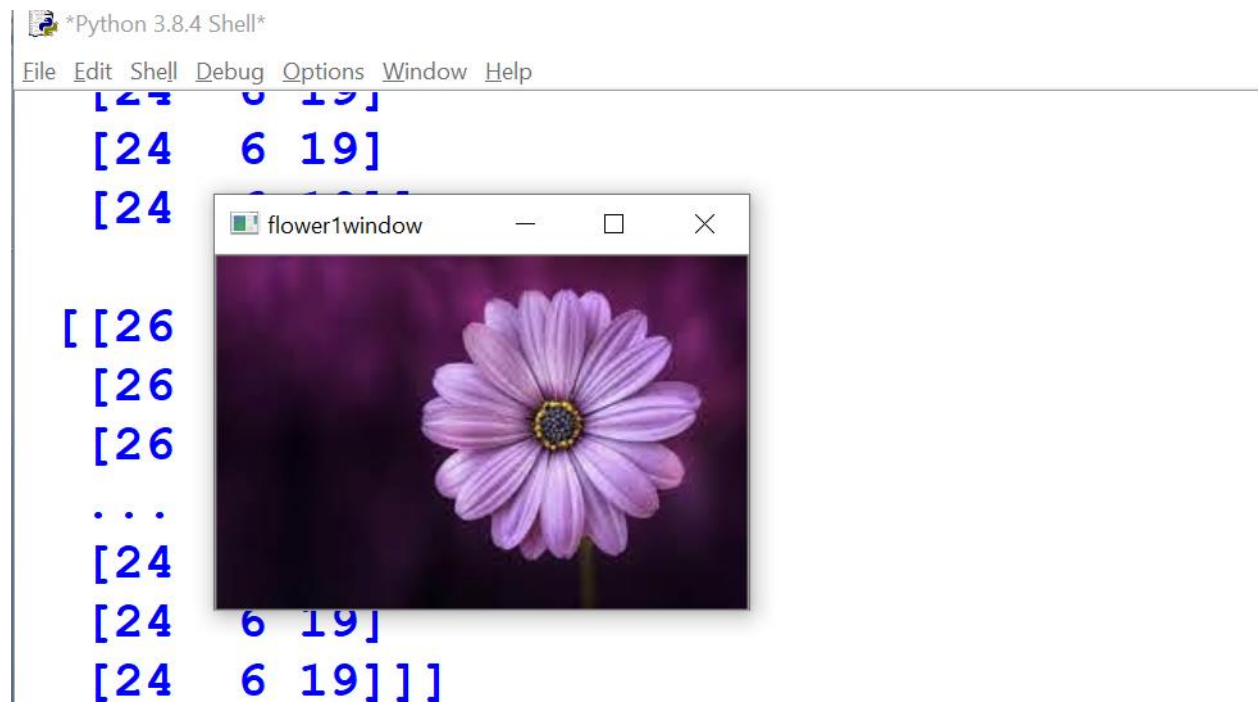
Output: image will display for 5min and gets closed

Step 11: Change flag to 0, it will give black and white image in the output

.

```
#read the image in matrix form
img = cv2.imread('flower1.jpg',0)
```

Step 12: If I make 5000 to 0, image will not get close until I click on x



Step 13: Let us improvise code to download and store image in our directory

```
opencvimage1.py - C:/IBMCE_Python/OpenCV/opencvimage1.py (3.8.4)
File Edit Format Run Options Window Help

#opencv example1
import cv2



#read the image in matrix form
img = cv2.imread('flower1.jpg',1)

cv2.imshow("flower1window",img)

k = cv2.waitKey(0)
#unicode for escape button is 27
if k==27:
    cv2.destroyAllWindows()
elif k==ord('s'):
    cv2.imwrite('flower11.jpg',img)
cv2.destroyAllWindows()

print(img)
```

Note: press 's' button to save the file

	flower1	01-12-2020 01:05
	flower11	01-12-2020 01:22

Let us work with real camera

Step 14: Open a new file with name 'videosample.py'

```
videosample.py - C:/IBMCE_Python/OpenCV/videosample.py (3.8.4)
File Edit Format Run Options Window Help
#video sample code
import cv2

cap=cv2.VideoCapture(0)

print(cap.isOpened())
```

Step 15: Execute and in output we should see True as we passed 0 as parameter

```
===== RESTART: C:/IBMCE_Python/OpenCV/videosample.py =
True
>>>
```

Step 16: put parameter as 1 and check output it should show false

```
===== RESTART: C:/IBMCE_Python/OpenCV/videosample.py =
False
>>>
```

Step 17: Add few more lines of code to work with camera

```
videosample.py - C:/IBMCE_Python/OpenCV/videosample.py (3.8.4)
File Edit Format Run Options Window Help
#video sample code
import cv2

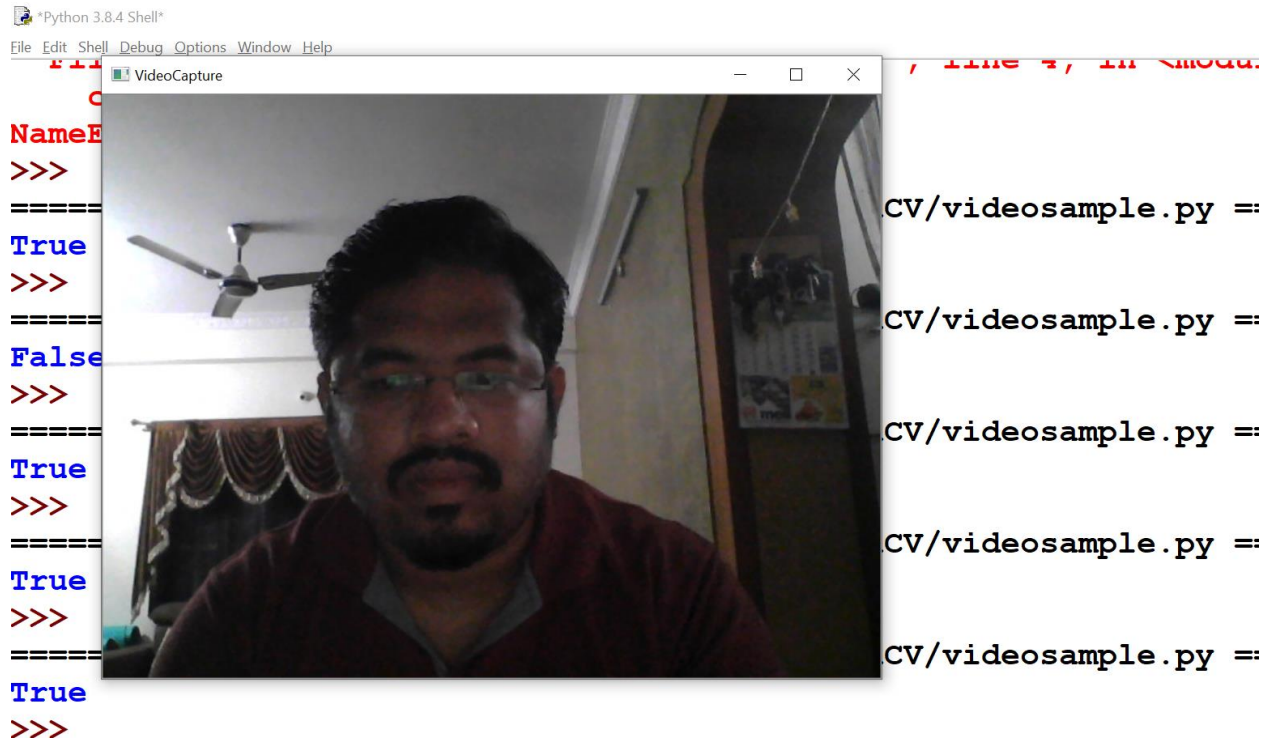
cap=cv2.VideoCapture(0)

print(cap.isOpened())

while(cap.isOpened()):
    ret, frame=cap.read()
    cv2.imshow("VideoCapture", frame)

    if cv2.waitKey(1) & 0xFF==ord('q'):
        break

#camera release
cap.release()
cv2.destroyAllWindows()
```

Clicking on q, it will get closed

Step 18: Now let us work with video writing (total 3 more lines of code we are writing)

```
videosample.py - C:/IBMCE_Python/OpenCV/videosample.py (3.8.4)
File Edit Format Run Options Window Help
#video sample code
import cv2

cap=cv2.VideoCapture(0)

rec=cv2.VideoWriter_fourcc(*'XVID')

out=cv2.VideoWriter('myvideo.avi',rec,20.0,(640,480))
#videofile,rec var,frame/sec, framesize

print(cap.isOpened())

while(cap.isOpened()):
    ret,frame=cap.read()

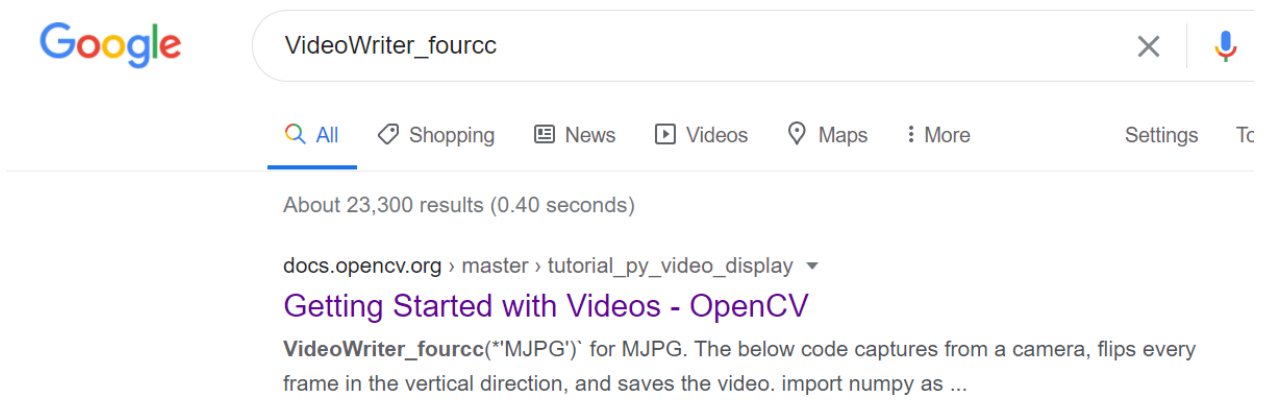
    #it will record and saved
    out.write(frame)

    cv2.imshow("VideoCapture",frame)







    if cv2.waitKey(1) & 0xFF==ord('q'):
        break

#camera release
cap.release()
cv2.destroyAllWindows()
```

Step 19: Know more about VideoWriter_fourcc by getting into URL



Step 20: Execute the code so that you can see a video file gets saved into the directory with name 'myvideo'

 flower1	01-12-2020 01:05	JPG File	5 KB
 flower11	01-12-2020 01:22	JPG File	11 KB
 myvideo	01-12-2020 01:55	AVI File	0 KB
 OpenCV_Lab	01-12-2020 01:22	Microsoft Office Wor...	1,065 KB
 opencvimage1	01-12-2020 01:28	Python File	1 KB
 videosample	01-12-2020 01:55	Python File	1 KB

Step 21: face.py file, always updates the same file 'facenow.jpg'

```
#face detection
import cv2
import numpy as np
import datetime
#these two are models built with lots of images
face_classifier=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
eye_classifier=cv2.CascadeClassifier("haarcascade_eye.xml")

#It will read the first frame/image of the video
video=cv2.VideoCapture(0)

while True:
    #capture the first frame
    check,frame=video.read()#black&white
    gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)#making into color

    #detect the faces & eyes from the video using detectMultiScale function
    faces=face_classifier.detectMultiScale(gray,1.3,5)#returns 4 values
    eyes=eye_classifier.detectMultiScale(gray,1.3,5) #returns 4 values

    print(faces)#matrix form displaying

    #drawing rectangle boundaries for the detected face
    for(x,y,w,h) in faces:
        #x and y axis of rectangle, RGB values,
        cv2.rectangle(frame, (x,y), (x+w,y+h), (127,0,255), 2)
        cv2.imshow('Face detection', frame)
        cv2.imwrite('facenow.jpg',frame)
        #picname = datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
        #cv2.imwrite(picname+".jpg", frame)
```

22. Now modify the code and import datetime module so that always it will be saved into a new file

```

#face detection
import cv2
import numpy as np
import datetime
#these two are models built with lots of images
face_classifier=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
eye_classifier=cv2.CascadeClassifier("haarcascade_eye.xml")

#It will read the first frame/image of the video
video=cv2.VideoCapture(0)
while True:
    #capture the first frame
    check,frame=video.read()#black&white
    gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)#making into color

    #detect the faces & eyes from the video using detectMultiScale function
    faces=face_classifier.detectMultiScale(gray,1.3,5)#returns 4 values
    eyes=eye_classifier.detectMultiScale(gray,1.3,5) #returns 4 values









    print(faces)#matrix form displaying

    #drawing rectangle boundaries for the detected face
    for(x,y,w,h) in faces:
        #x and y axis of rectangle, RGB values,
        cv2.rectangle(frame, (x,y), (x+w,y+h), (127,0,255), 2)
        cv2.imshow('Face detection', frame)
        #cv2.imwrite('facenow.jpg',frame)
        picname = datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
        cv2.imwrite(picname+".jpg",frame)
    #Note: if we rerun it will update the same file

```

Output:

> This PC > Local Disk (C:) > IBMCE_Python > facedetection

	Name	Date modified	Type	Size
★	 20-11-29-11-31	29-11-2020 11:31	JPG File	73 KB
★	 20-11-29-11-32	29-11-2020 11:32	JPG File	74 KB
★	 20-12-01-02-08	01-12-2020 02:08	JPG File	68 KB
★	 20-12-01-02-16	01-12-2020 02:16	JPG File	69 KB
★	 20-12-01-02-17	01-12-2020 02:17	JPG File	64 KB
	 20-12-01-02-23	01-12-2020 02:23	JPG File	66 KB
	 20-12-01-02-33	01-12-2020 02:33	JPG File	69 KB
	 20-12-01-02-33	01-12-2020 02:33	JPG File	69 KB