

# Day 1: Understanding the basics

## Outcomes:

- Be able to capture video from a camera
- Understand what a frame is
- Know how pixels and arrays represent images
- Display frames in color and in grayscale

# What is a camera?

A camera is a machine that gives you images very fast.

Video = image + image + image + image + ...

(About 30 images per second)

Each image is called a **Frame**.

## Frames

Frame = Matrix of Pixels = A NumPy Array

### Frames as a Matrix of Pixels

You can imagine Frame as the **Matrix of Pixels** as:

A 2 x 2 Frame:

Pixel 1 (0,0) [B,G,R]	Pixel 2 (0,1) [255, 0, 0]
Pixel 3 (1,0) [0, 0, 255]	Pixel 4 (1,1) [255, 255, 255]

## Frames as a NumPy array:

(Machine view)

Shape of the NumPy array (for color images):

```
(height, width, 3)
```

Height → number of rows of pixels

Width → number of columns of pixels

3 → Blue, Green, Red channels (BGR)

A 2 x 2 Frame as a NumPy Array:

```
frame = [  
    [ [B,G,R],      [0,200,50] ],    # row 0  
    [ [255,0,0],     [0,0,0]   ]      # row 1  
]
```

# What is Motion Tracking?

Motion Tracking = Comparing frames over time

Before Tracking, we need to learn to:

- Open the camera
- Read the frames
- Display them

## Installing OpenCV

- Open Terminal and type:

```
pip install opencv-python
```

Hit enter and OpenCV should be installed.

# Opening the camera

Create a new file.

1-camera.py:

```
import cv2

# Open the default webcam
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read() #read one frame
    if not ret:
        break

    cv2.imshow("Webcam", frame) #show frame

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

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

## Understanding the code:

<code>cv2.VideoCapture(0)</code>	Opens your webcam
<code>cap.read()</code>	Reads one frame → Returns a matrix of pixels
<code>cv2.imshow()</code>	Displays the frame in the window
<code>cv2.waitKey(1)</code>	Waits 1ms for a key press
<code>cap.release()</code>	Closes the webcam
<code>cv2.destroyAllWindows</code>	Closes the window

(Explained in detail in Day 2)

## Grayscale view

In 1-camera.py

change:

```
cv2.imshow("Camera", frame)
```

to:

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
cv2.imshow("Camera", gray)
```

and run the code.

Note that cv2.cvtColor converts color spaces from **BGR to Gray**.

(optional):

2-grayscale.py

```
import cv2

cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    if not ret:
        break

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    cv2.imshow("Camera", gray)

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

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