



INDIAN
INSTITUTE OF
TECHNOLOGY
BACHULAL SWAMI NAYANAR



LARSEN & TOUBRO

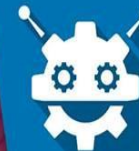
PRESENTS



THE ANNUAL TECHNO-MANAGEMENT FEST, IIT (BHU) VARANASI

TECHNEX'17

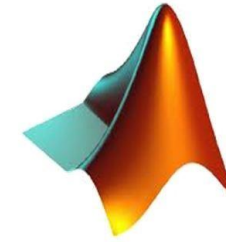
TOWARDS SUSTAINABILITY • 24-25 FEBRUARY 2017



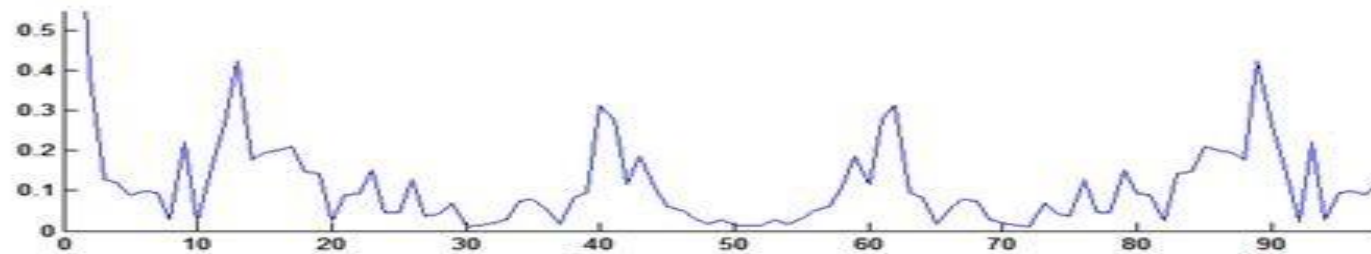
PIXELATE

THE ROBOT EYE





Workshop on Video Processing using MATLAB



MOTIVATION

- Face Recognition
- Gesture Recognition
- Object follower
- Machine Learning

The scope of this workshop

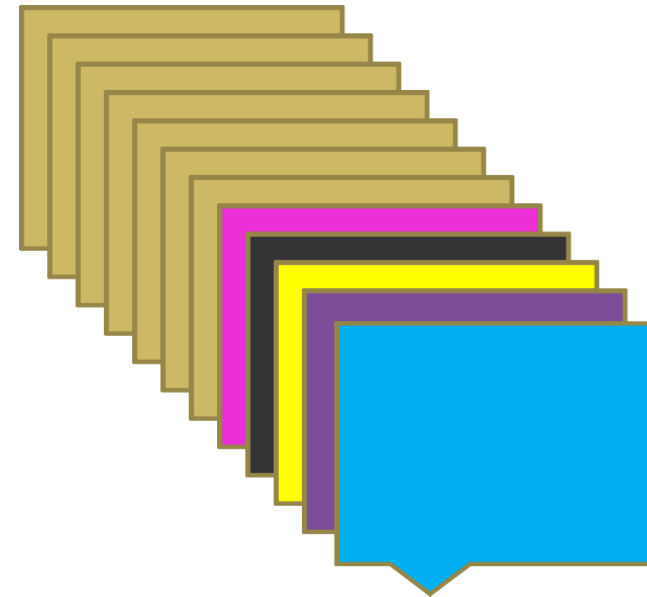
1. What is video processing
2. How to record a video
3. How to take video input in real time
4. How do I see what is being recorded
5. How to work on a video
6. How is a video stored in workspace
7. How to choose the length of a video
8. How to choose the speed of camera taking a video
9. How can I save my video in my computer
10. How to do processing on a video

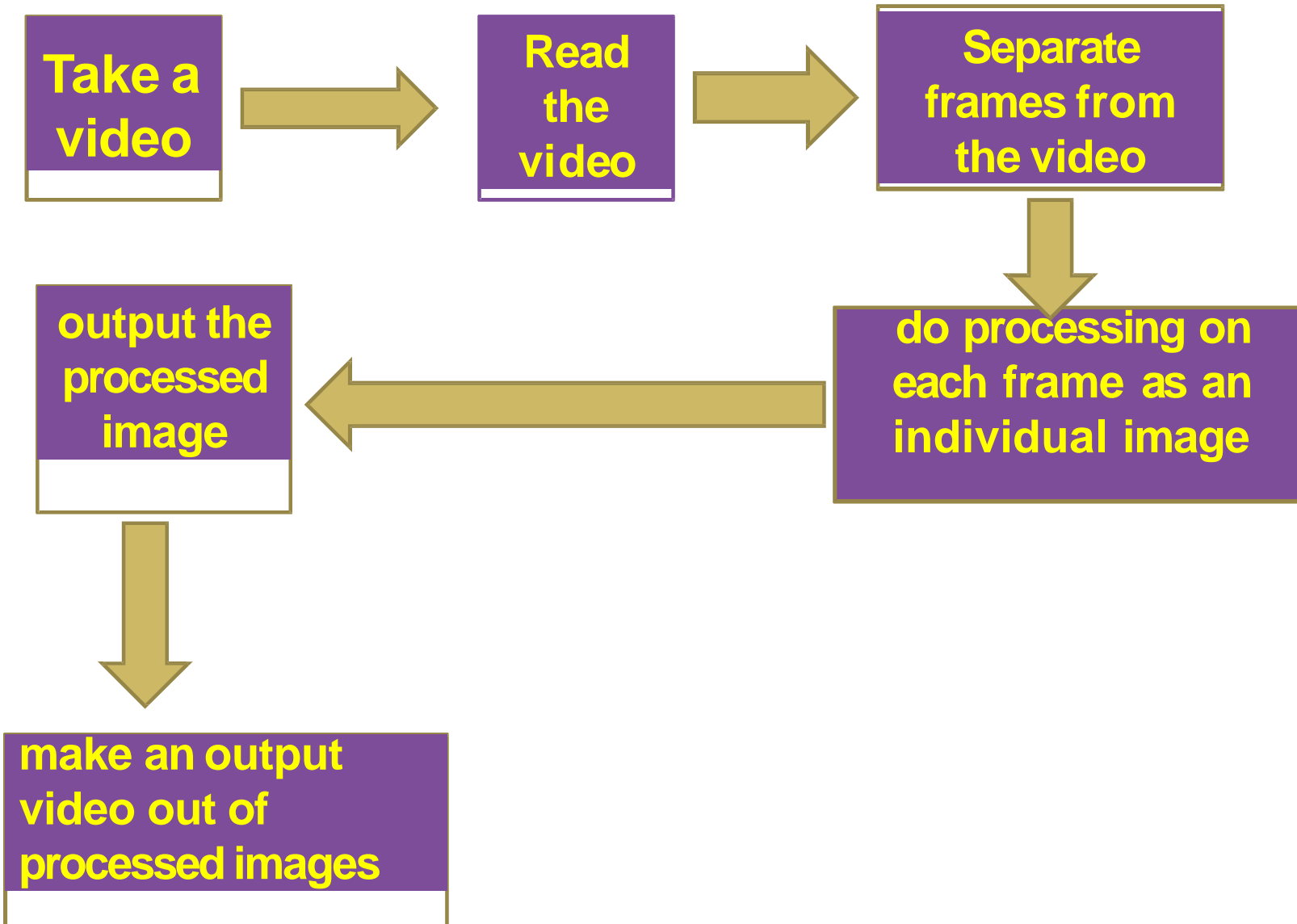
❑ A video is actually a 4-D variable in matlab workspace.

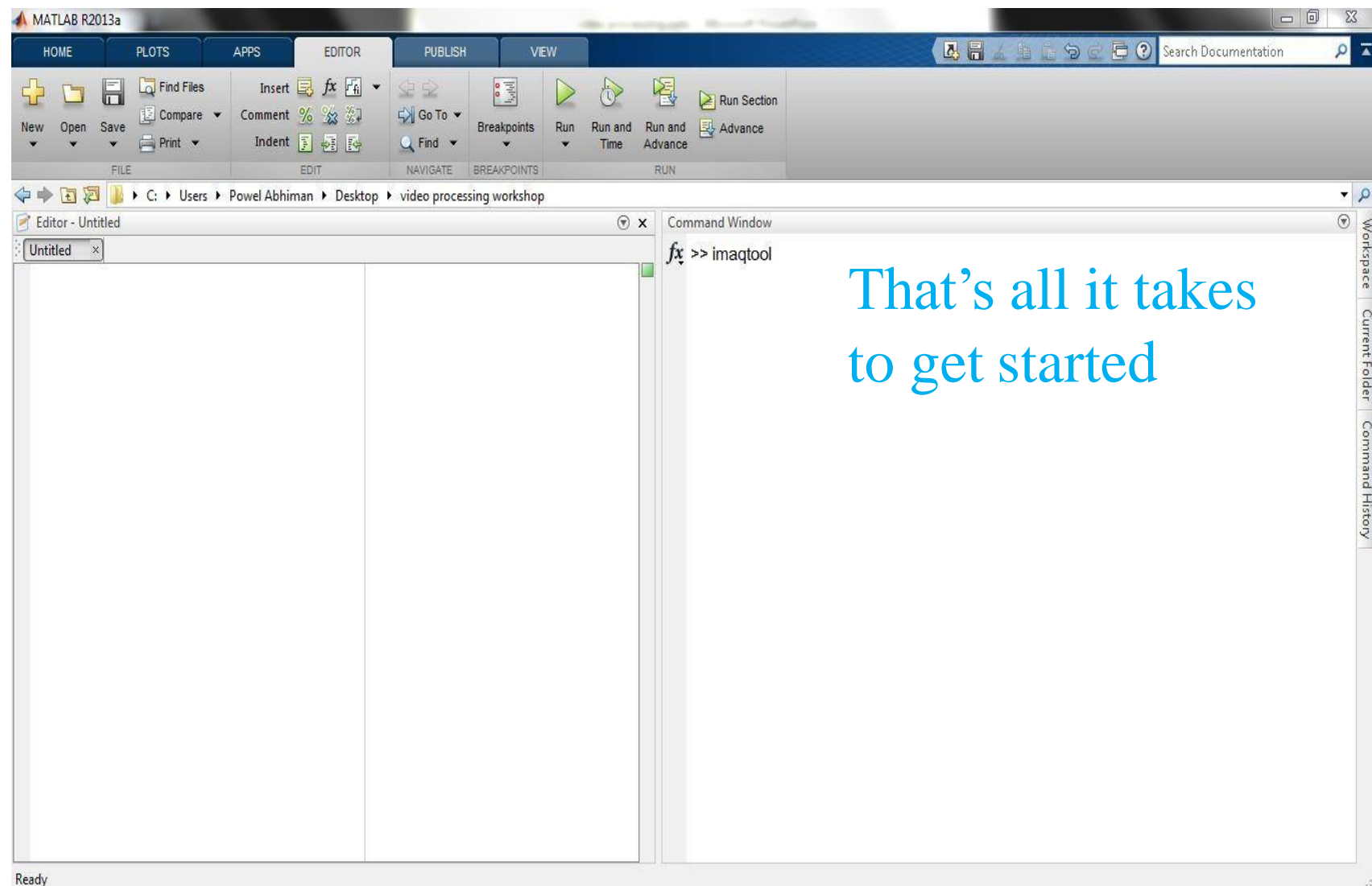
❑ The 4th dimension represents the no. of frames.

❑ Frames can be RGB image or Grayscale or binary image.

❑ Videos are extension of images in 4th dimension.







That's all it takes
to get started

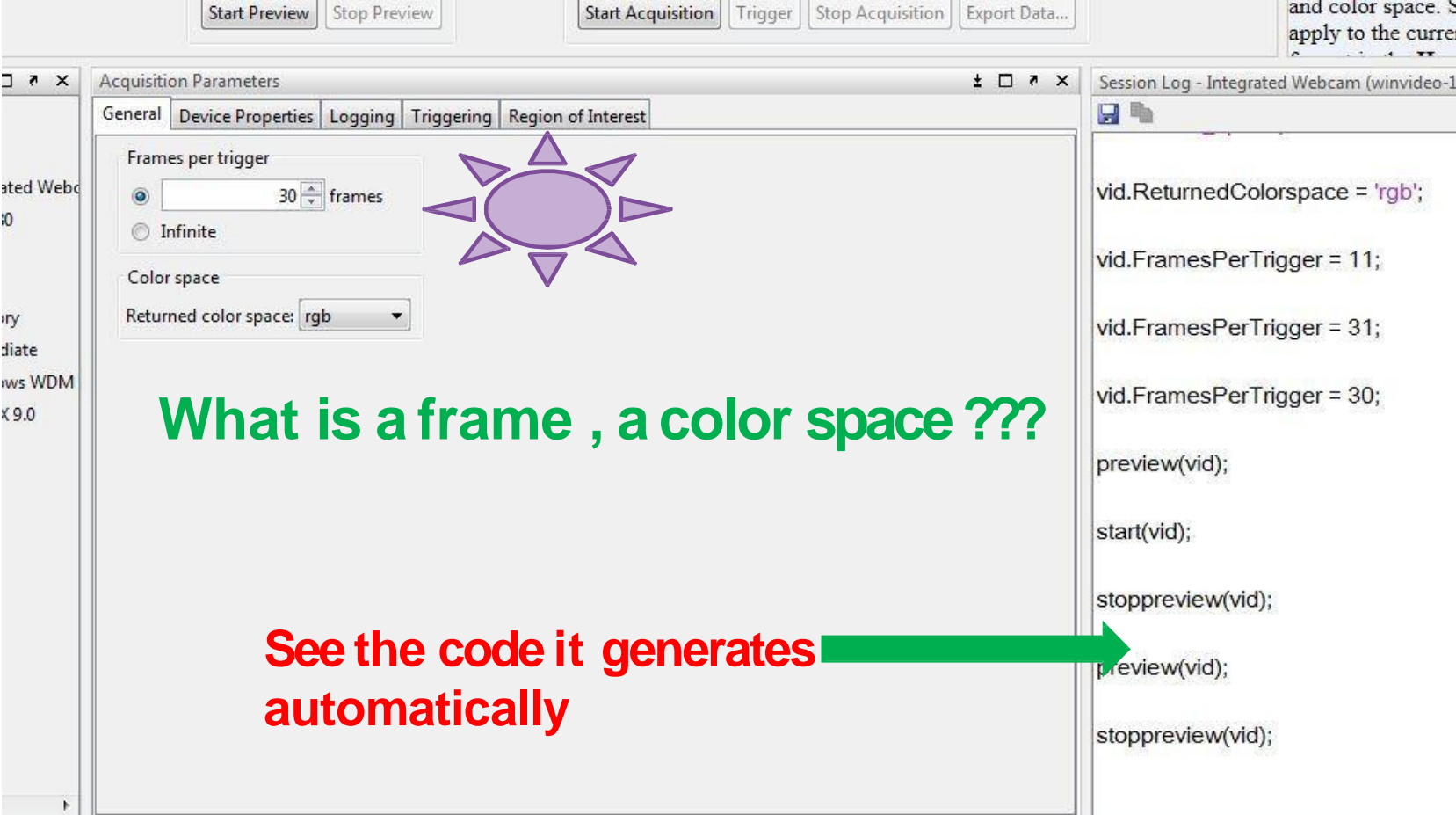
Image acquisition toolbox

The screenshot shows the 'Image Acquisition Tool' window with several panes and annotations:

- Hardware Browser:** A tree view on the left showing connected devices and their supported formats. A red box highlights the 'Integrated Webcam (winvideo-1)' node, with the text 'Select ur camera resolution here' in red.
- Preview window:** A large central area for viewing the camera feed. It contains the text 'Preview window' in green and 'Find Help here' in blue. A purple arrow points from the 'Find Help here' text to the 'Desktop Help' pane.
- Desktop Help:** A pane on the right containing text about the Hardware Browser and acquisition process. It includes instructions like 'click Tools > Refresh Image Acquisition Hardware' and 'Select the device format or camera file you want to use for the acquisition'.
- Acquisition Parameters:** A pane at the bottom center for configuring acquisition settings. It contains the text 'Alter ur acquisition parameters here' in green. A purple arrow points from this text to the 'Information' pane.
- Information:** A pane at the bottom left showing software details like 'Toolbox version: 4.5 (R2013a)' and 'MATLAB version: 8.1 (R2013a)'. It contains the text 'Info about ur device comes here' in purple.
- Session Log:** A pane at the bottom right for viewing the acquisition log. It contains the text 'Ur matlab generated code will come here' in blue.

Buttons at the bottom include 'Start Acquisition', 'Trigger', 'Stop Acquisition', 'Export Data...', 'Start Preview', and 'Stop Preview'.

General Properties



The screenshot shows the 'Acquisition Parameters' dialog box with the 'General' tab selected. The 'Frames per trigger' section has a radio button selected for '30 frames' and another for 'Infinite'. The 'Color space' section has a dropdown menu set to 'rgb'. A purple sun icon is drawn over the dialog box. Overlaid on the dialog box is the text 'What is a frame , a color space ???' in green. Below this, in red, is the text 'See the code it generates automatically' with a green arrow pointing to the right, towards the 'Session Log' window. The 'Session Log' window on the right shows a list of commands: 'vid.ReturnedColorspace = 'rgb';', 'vid.FramesPerTrigger = 11;', 'vid.FramesPerTrigger = 31;', 'vid.FramesPerTrigger = 30;', 'preview(vid);', 'start(vid);', 'stoppreview(vid);', 'preview(vid);', and 'stoppreview(vid);'. The 'Start Preview' button is highlighted in the top bar of the dialog box.

Start Preview Stop Preview Start Acquisition Trigger Stop Acquisition Export Data...

Acquisition Parameters

General Device Properties Logging Triggering Region of Interest

Frames per trigger

☒ 30 frames

☐ Infinite

Color space

Returned color space: rgb

What is a frame , a color space ???

See the code it generates automatically

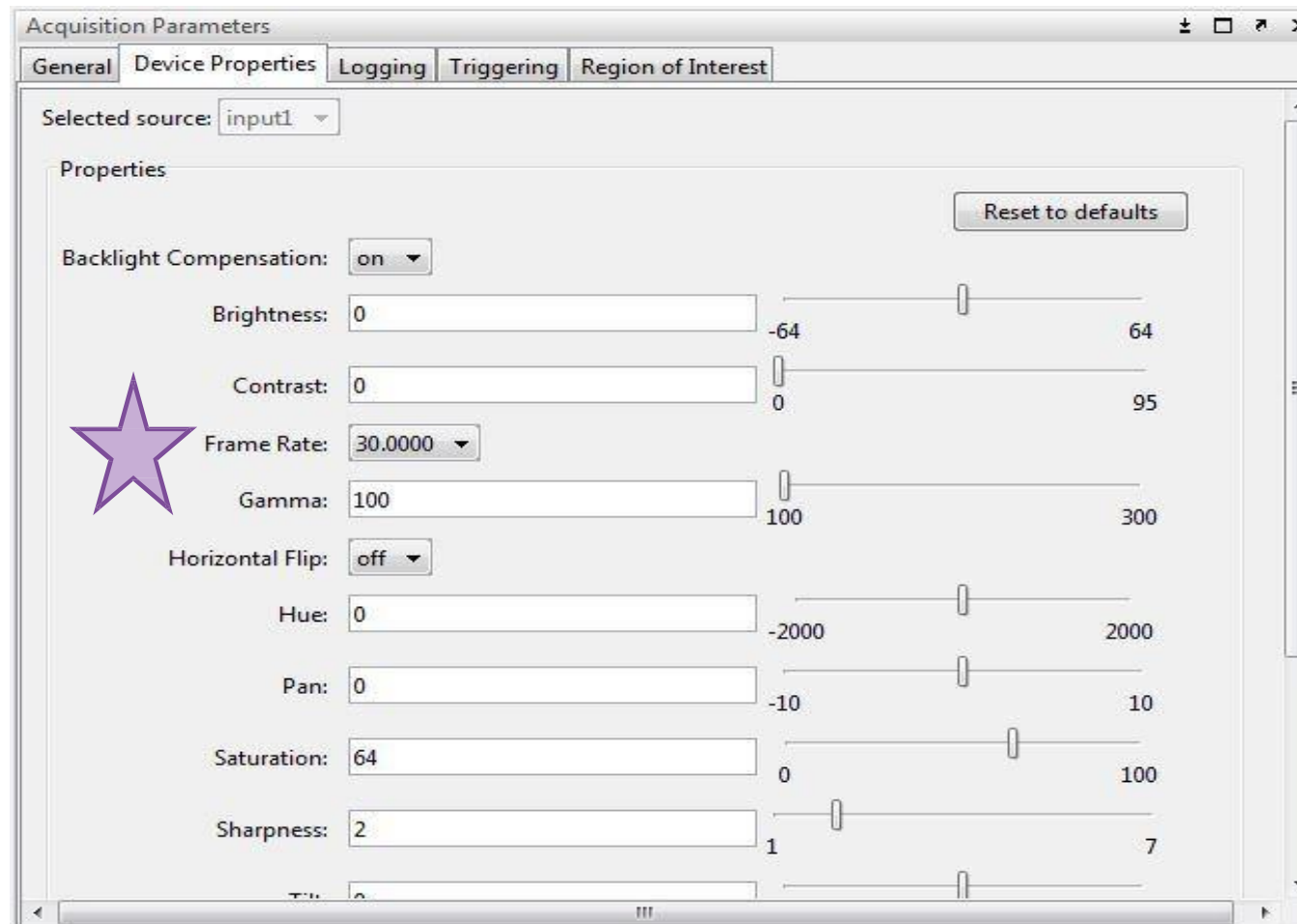
Session Log - Integrated Webcam (winvideo-1)

```
vid.ReturnedColorspace = 'rgb';
vid.FramesPerTrigger = 11;
vid.FramesPerTrigger = 31;
vid.FramesPerTrigger = 30;
preview(vid);
start(vid);
stoppreview(vid);
preview(vid);
stoppreview(vid);
```

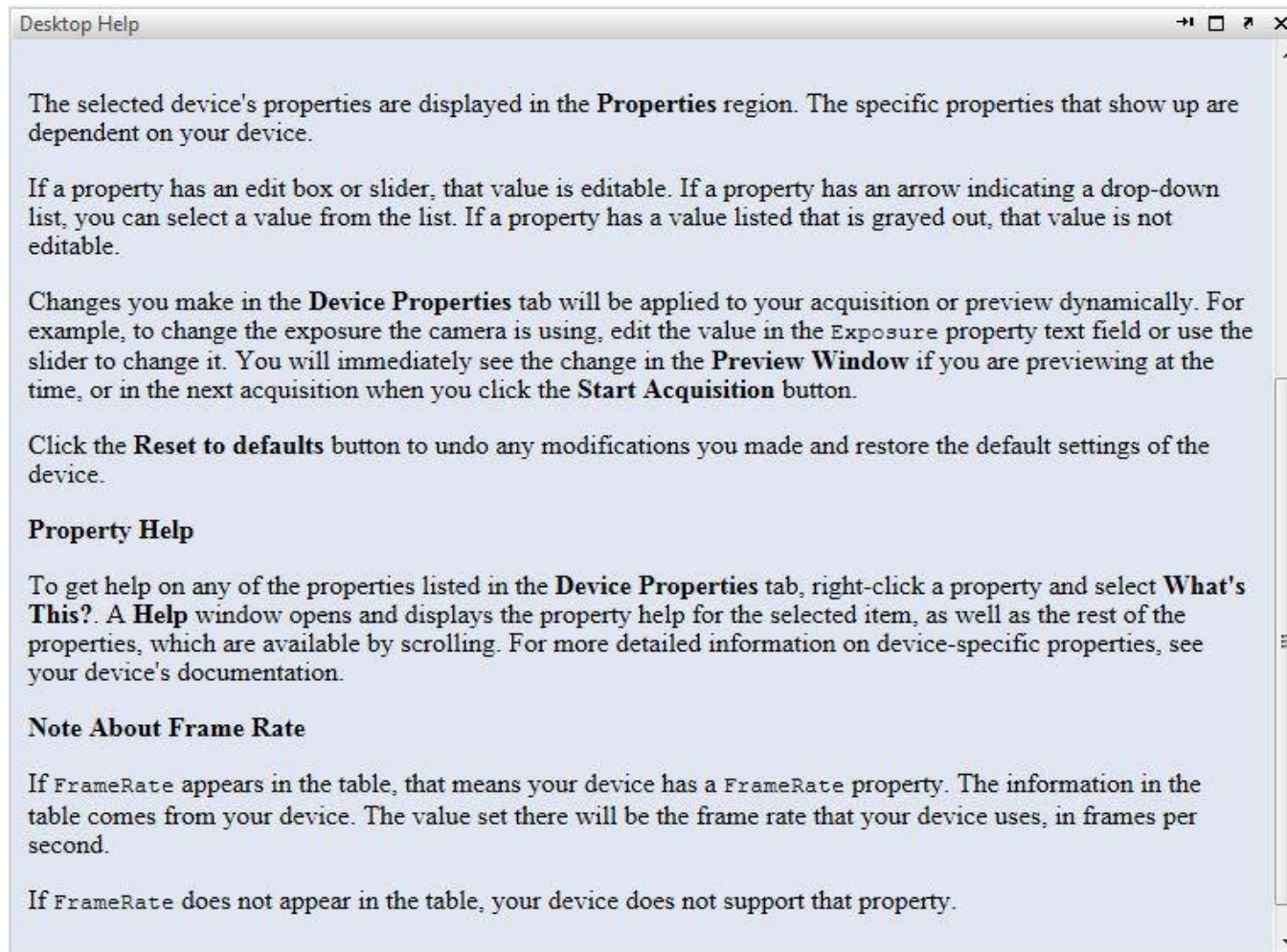
Help with Help



Setting your device properties



Help with Help, contd.

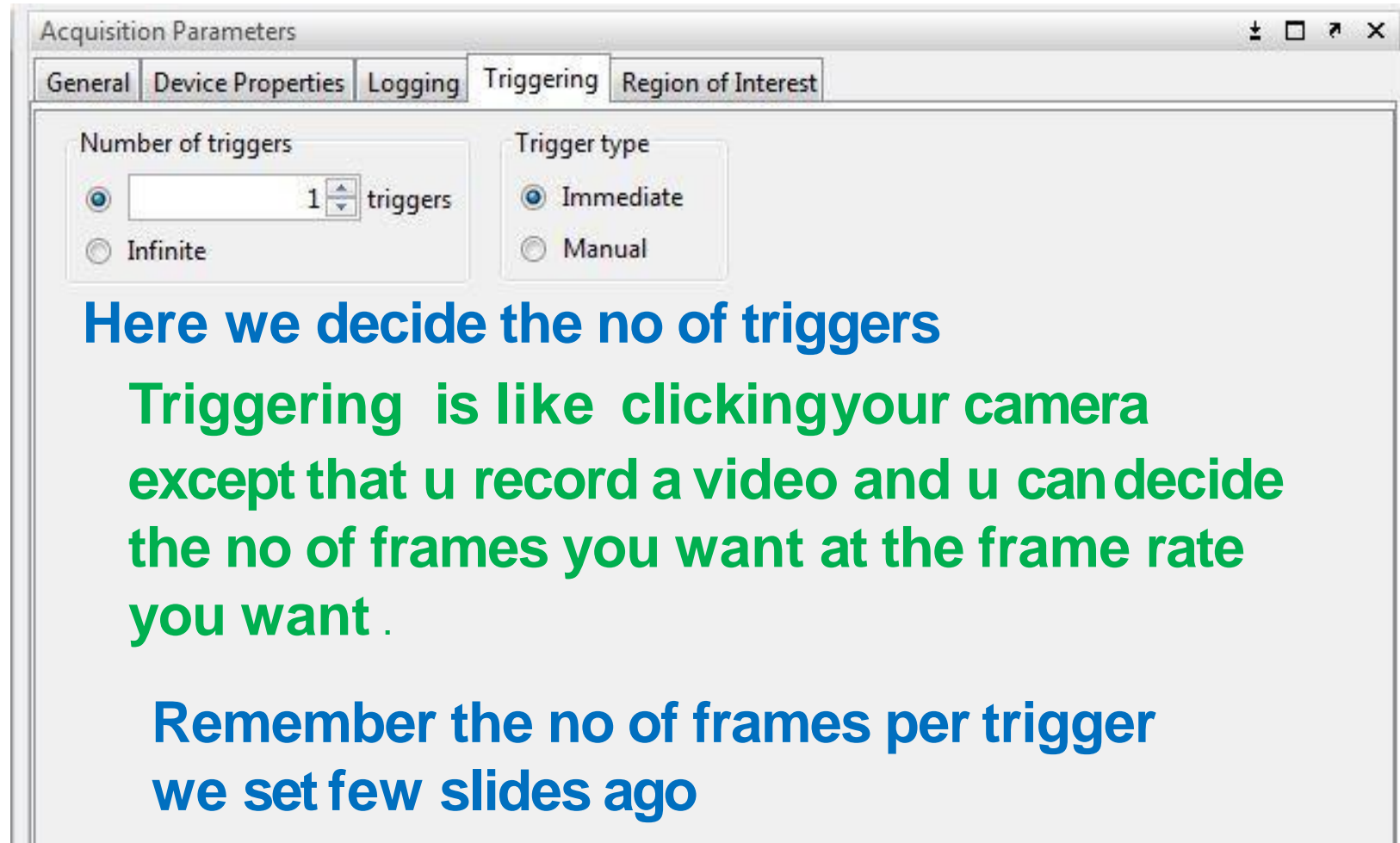


How and where to save your video ???

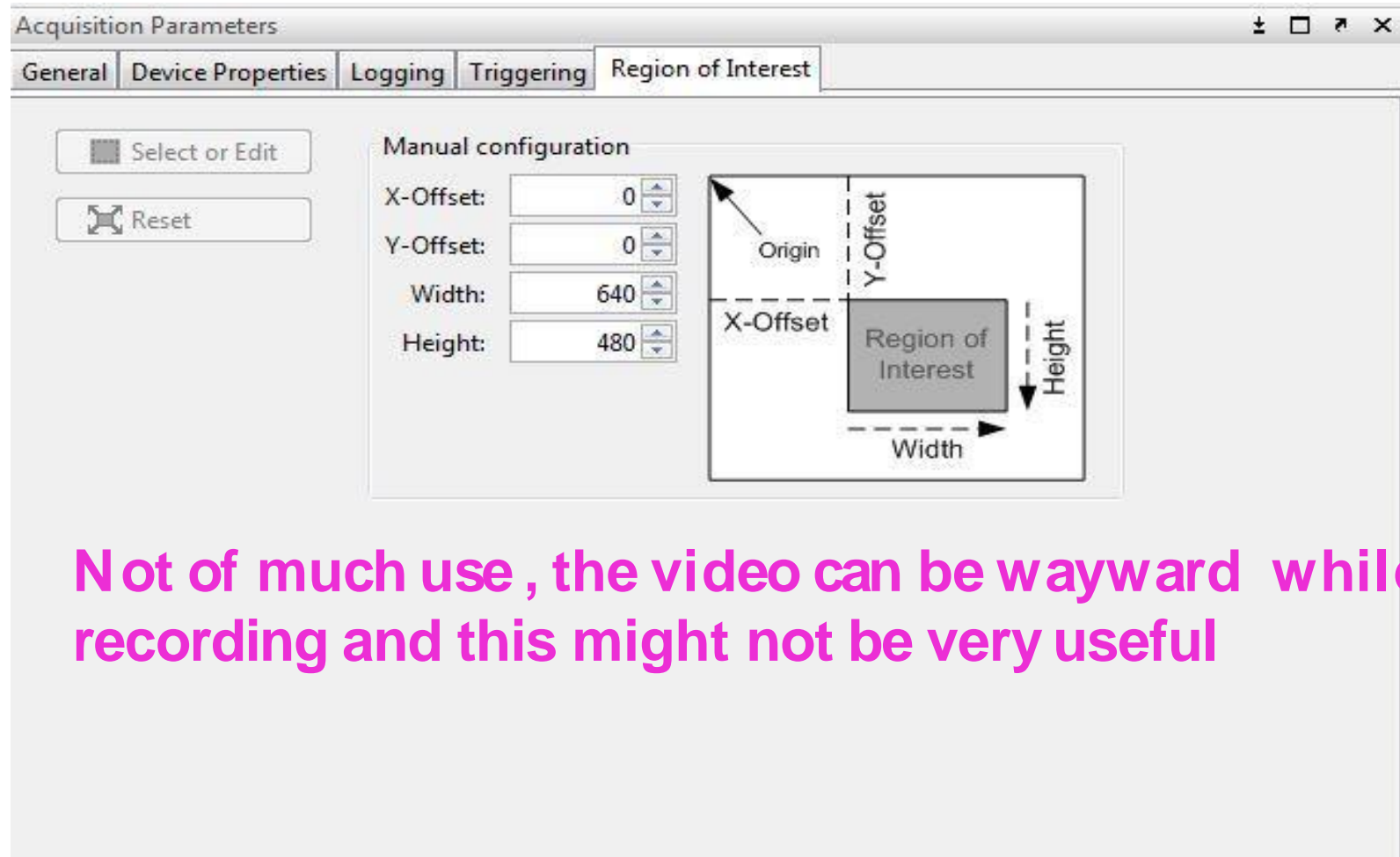
The screenshot shows a software window with five tabs: 'General', 'Device Properties', 'Logging', 'Triggering', and 'Region of Interest'. The 'Logging' tab is active. It contains two main sections: 'Log to' and 'Memory logging'. The 'Log to' section has three radio buttons: 'Memory' (selected), 'Disk', and 'Disk and memory'. The 'Memory logging' section has a 'Memory limit' field set to '1000.0 MB'. Below this is the 'Disk logging (VideoWriter)' section, which includes a 'Filename' field with a 'Browse...' button, an unchecked checkbox for 'Automatically increment filename', and a 'Profile' dropdown menu set to 'Uncompressed AVI'.

Memory = workspace and current folder as .mat file
Disk = current folder as .avi file for future use
.mat files cannot be played in vlc player, .avi can

Deciding the frame count



Select your region of interest

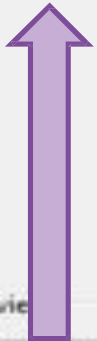


Not of much use , the video can be wayward while recording and this might not be very useful

Start viewing your video



**Start
preview**



- Remember previewing is different from actually acquiring the video as clear from the different options.

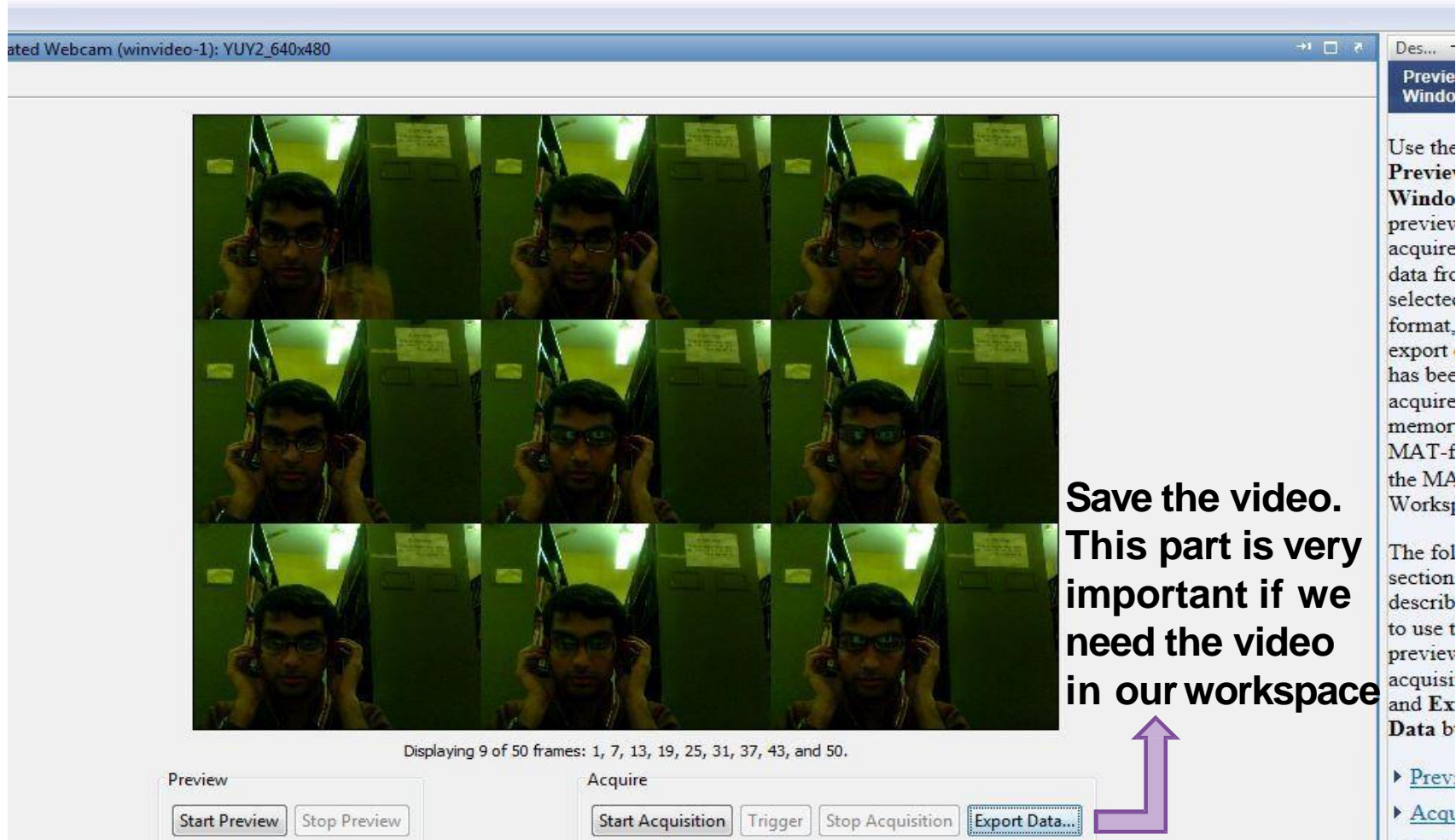
**Start
acquisition**



- How do the different options work

Recording the video

This is where the fun part starts



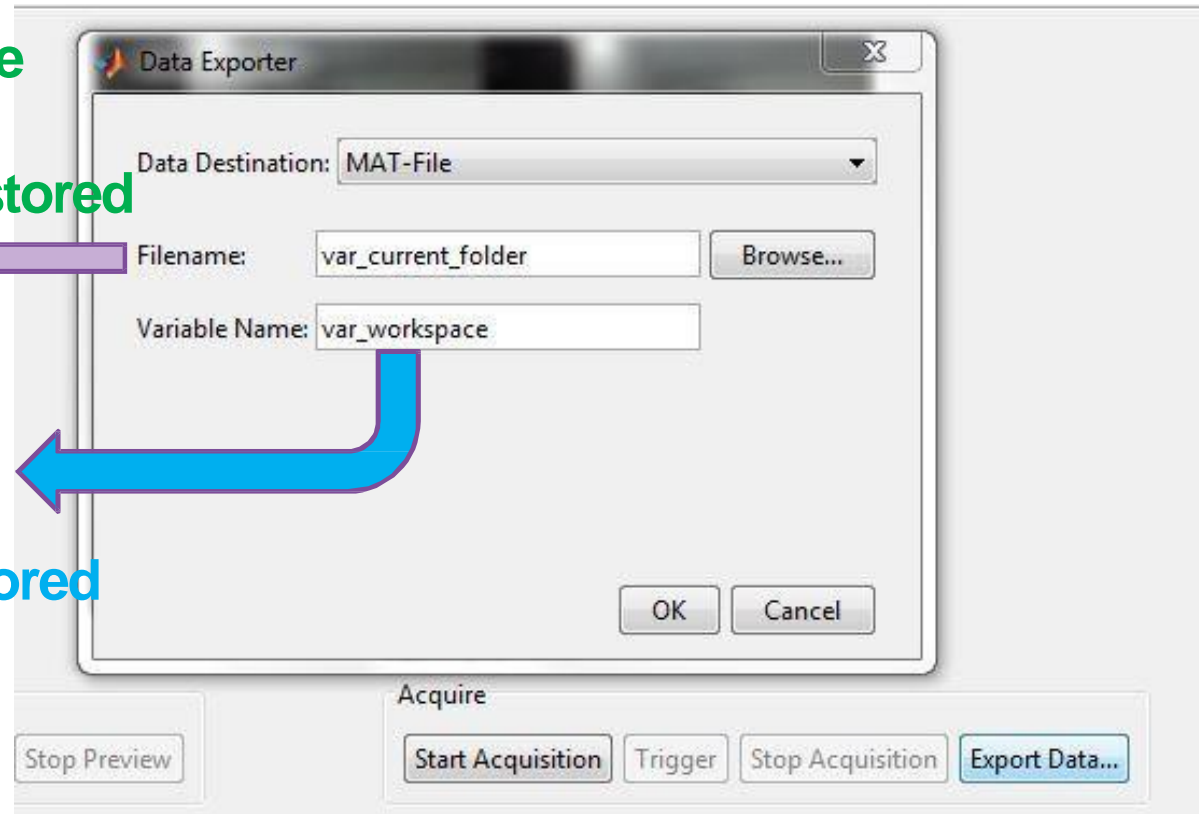
The screenshot shows a software window titled "ated Webcam (winvideo-1): YUY2_640x480". The main area displays a 3x3 grid of video frames showing a person wearing glasses and holding a phone to their ear. Below the grid, it says "Displaying 9 of 50 frames: 1, 7, 13, 19, 25, 31, 37, 43, and 50." At the bottom, there are two sections: "Preview" with "Start Preview" and "Stop Preview" buttons, and "Acquire" with "Start Acquisition", "Trigger", "Stop Acquisition", and "Export Data..." buttons. A purple arrow points from the "Export Data..." button to the text "Save the video. This part is very important if we need the video in our workspace". On the right side, there is a "Preview Window" panel with text about previewing and acquiring data, and a "Data" panel with "Preview" and "Acquire" links.

Save the video. This part is very important if we need the video in our workspace

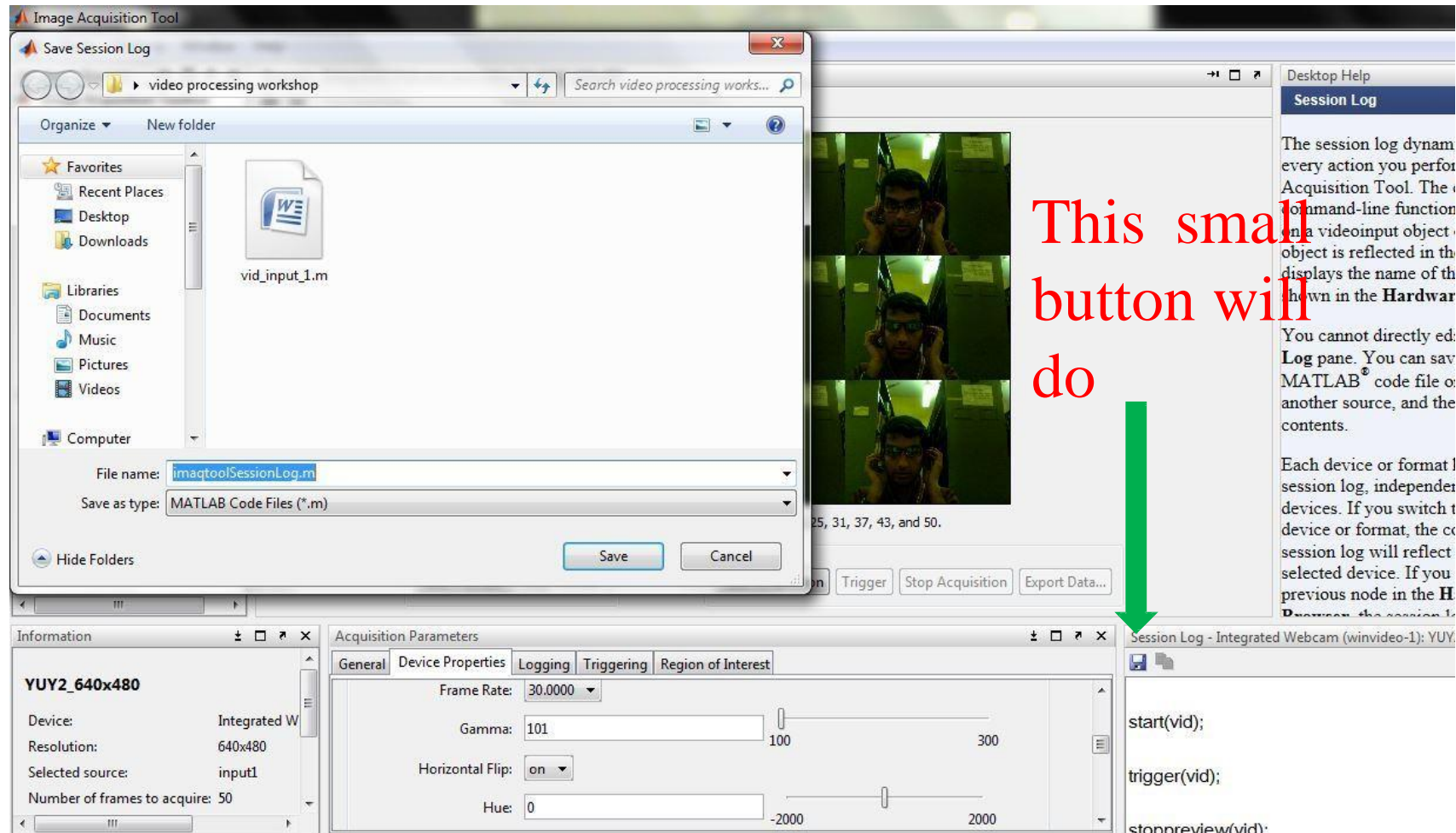
Actually saving the video data

This will be the
name
of variable as stored
in your
current folder

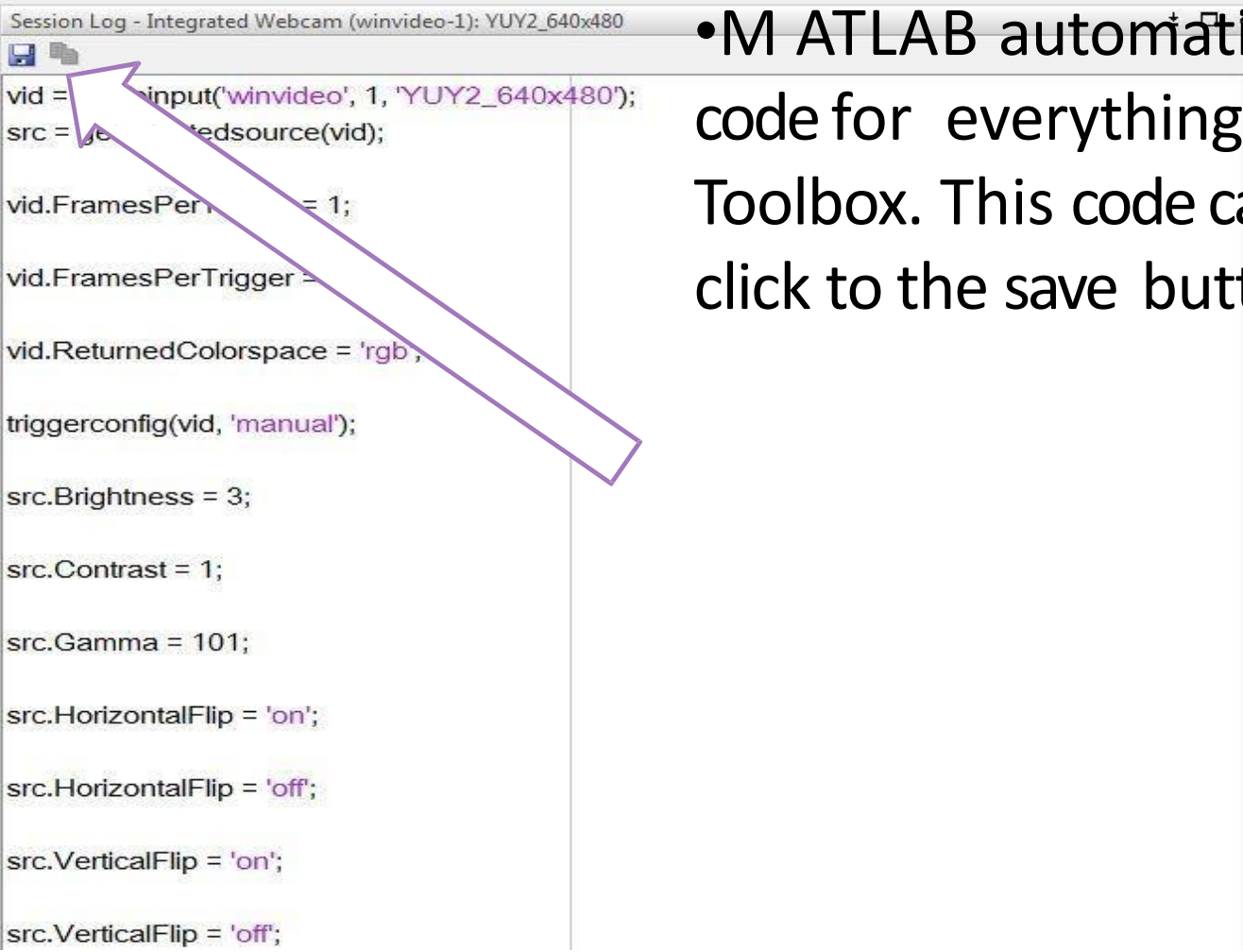
This will be the
name
of variable as stored
in your
workspace



Time to save our code



And we have a code 😊😊



A screenshot of a MATLAB 'Session Log' window titled 'Session Log - Integrated Webcam (winvideo-1): YUY2_640x480'. The window contains a list of MATLAB commands generated by the software. A purple arrow points from the text 'M ATLAB automatically generates the code for everything we do in the Toolbox' to the first line of code. The code is as follows:

```
vid = videoinput('winvideo', 1, 'YUY2_640x480');
src = getframe(source(vid));

vid.FramesPerTrigger = 1;

vid.FramesPerTrigger = 1;

vid.ReturnedColorspace = 'rgb';

triggerconfig(vid, 'manual');

src.Brightness = 3;

src.Contrast = 1;

src.Gamma = 101;

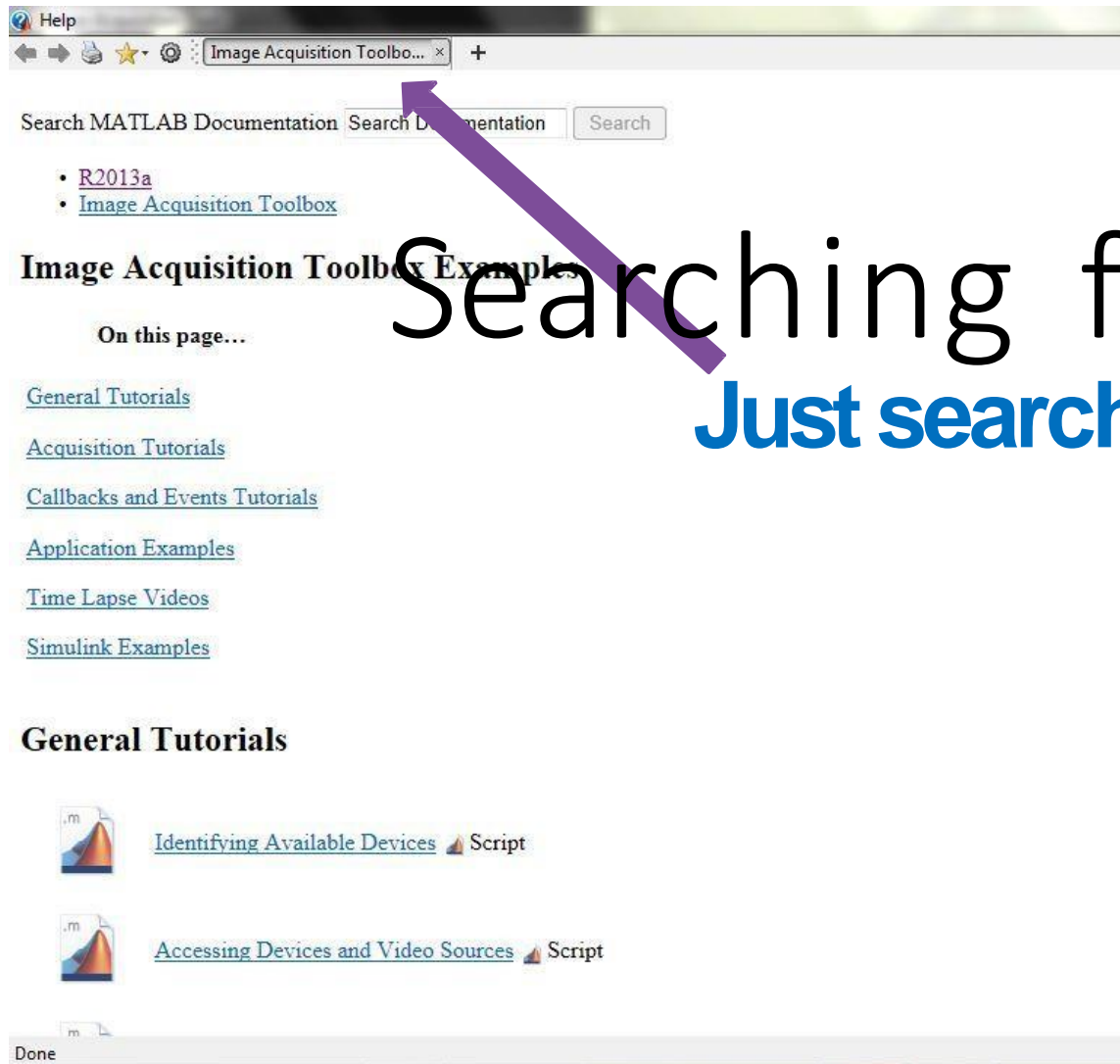
src.HorizontalFlip = 'on';

src.HorizontalFlip = 'off';

src.VerticalFlip = 'on';

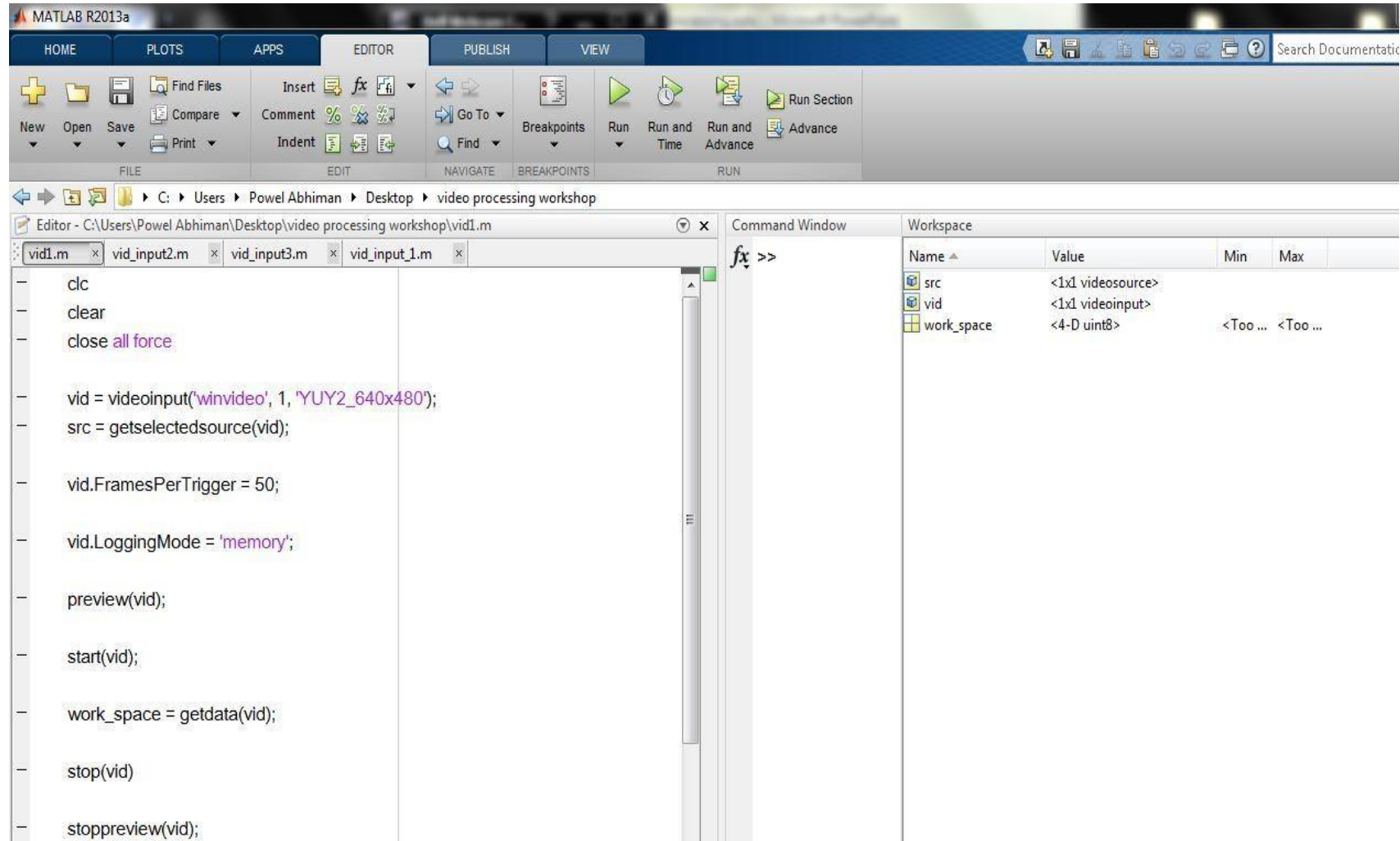
src.VerticalFlip = 'off';
```

- M ATLAB automatically generates the code for everything we do in the Toolbox. This code can be save just by one click to the save button.



Searching for help
Just search for help

The whole program



Video workspace stored in workspace as a 4-d array after the program is run.

How to take video input for the program

- Recorded videos from the computer eg '**vid1.wmv**'
vision.videoFileReader
- Record a video every time the program starts
using '**imaqtool**' at the start of program
- Tell the program to acquire frames when it requires
(**REAL TIME DATA**), '**getsnapshot()**'

Vid_input1.m & Vid_input2.m

What do I do with the frame?

- ❖ When we take a frame, all we need to do is some simple image processing.
- ❖ Video processing is just doing image processing on lots of frames of a video.

How to show output video from program

- Keep on storing the image frames in a 4-D array and play the whole data after the program is over.

imshow()

- Output the video as the frames are ready for display

videoWriter()

Vid_input1.m & Vid_input2.m

```
vid_input1.m x vid_input2.m x vid3.m x
%%
- clc
- close all force
- clear

%% read a video

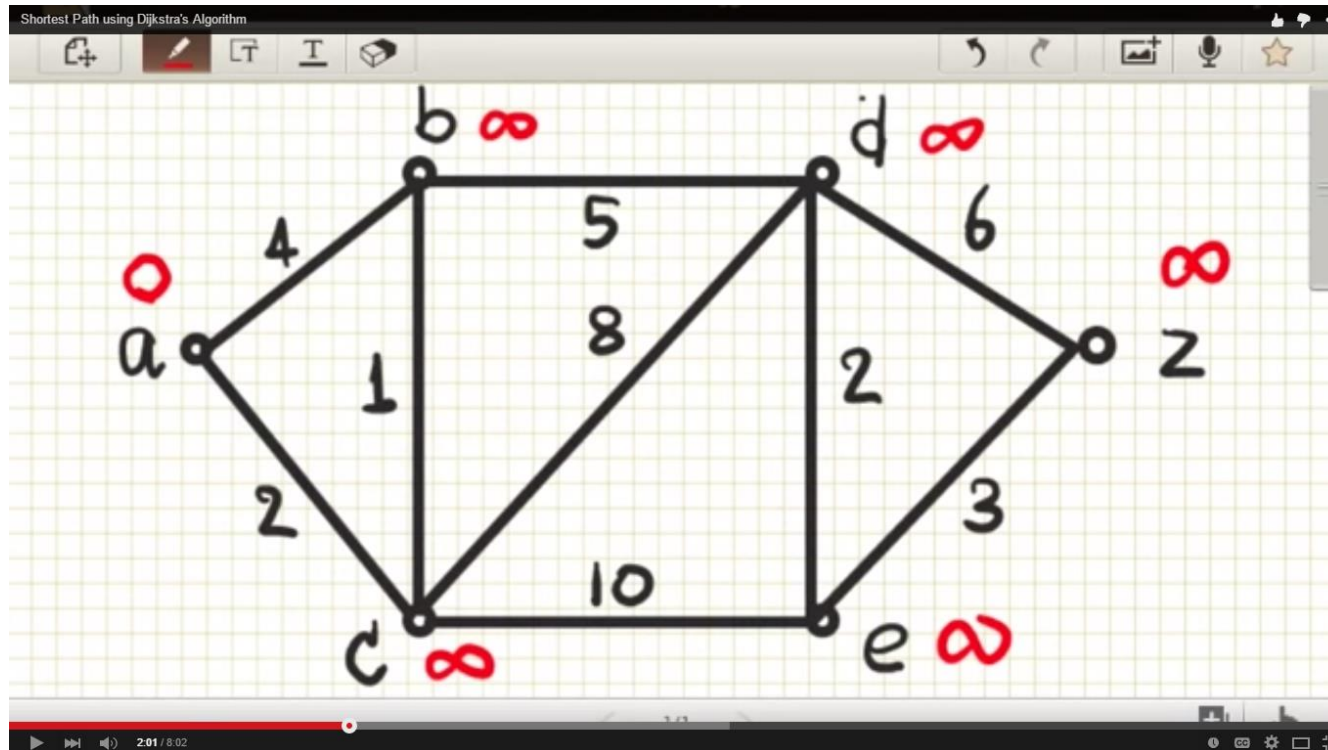
- videoFReader = vision.VideoFileReader('led0.wmv');
- videoPlayer = vision.VideoPlayer;

%% iterate over the frames
- while ~isDone(videoFReader)
-     frame = step(videoFReader);
-     im = frame(:, :, 1) > .250 & frame(:, :, 3) < .950 ;
-     step(videoPlayer, im);
- end
```


- 1. Are the input and output ways dependent on each other?**
- 2. Why cant we use 'imshow' to view the frames or processed images??**
- 3. How then, should we see the processed imaged for debugging??**
- 4. Why is getSnapShot() so slow ??**

- Get a video
- Extract frames
- Work on frames (Image processing)
- Output the resulting image
- Make/play final video

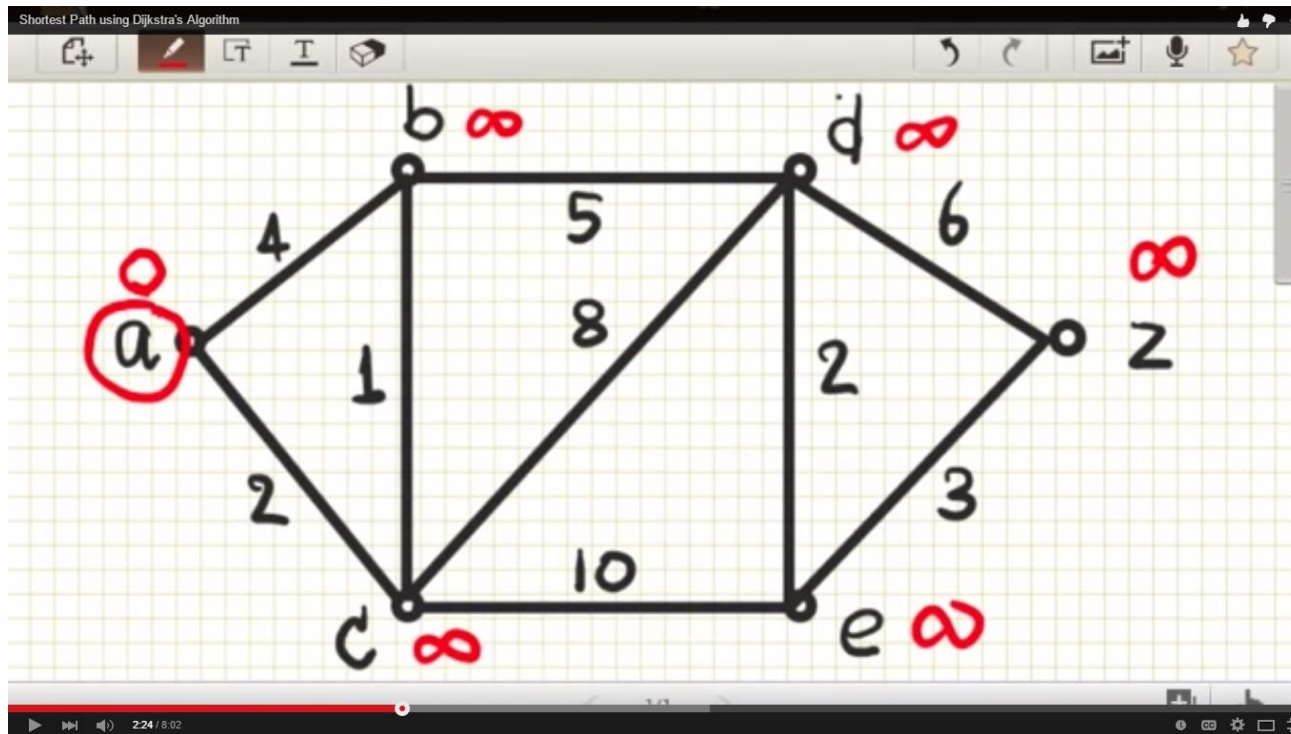
Dijkstra Algorithm



All nodes except initial node are initialized to Some random infinite distance.

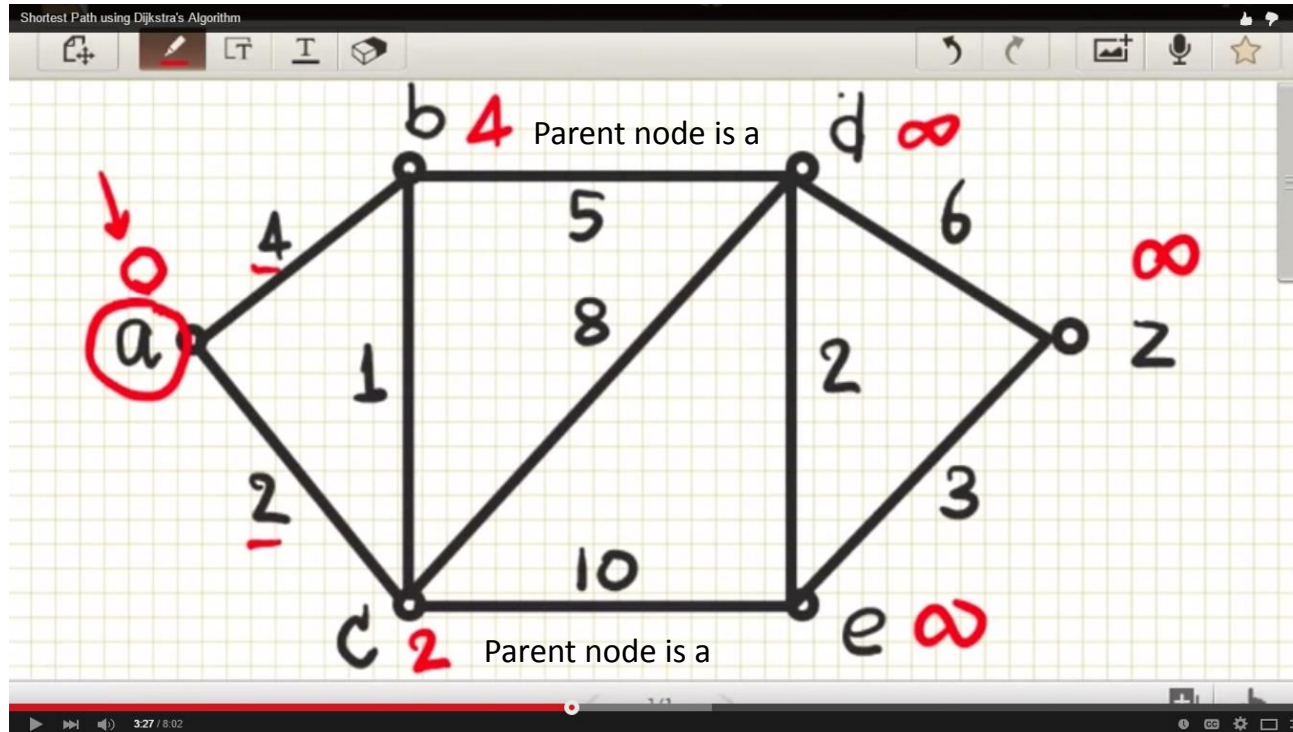
The node with the least distance is selected as the current node.

Here initial node becomes current node as it has a distance of zero.



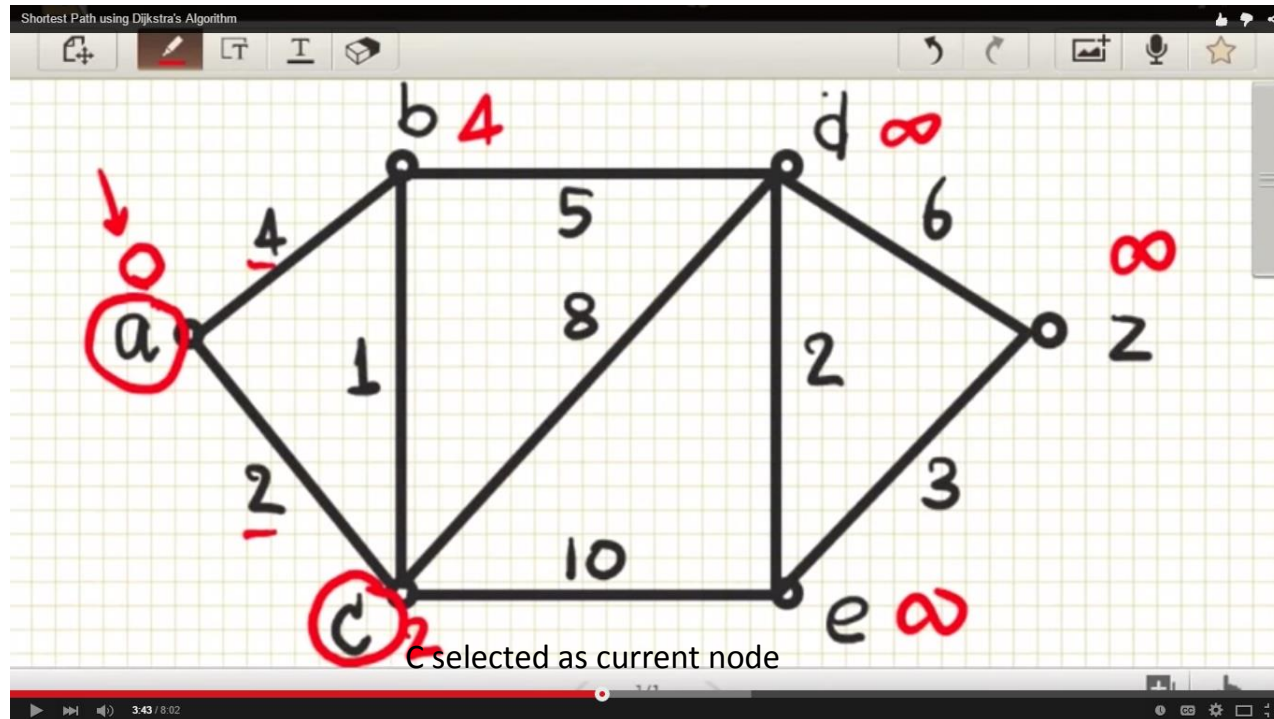
Here the initial node
Is set as current node

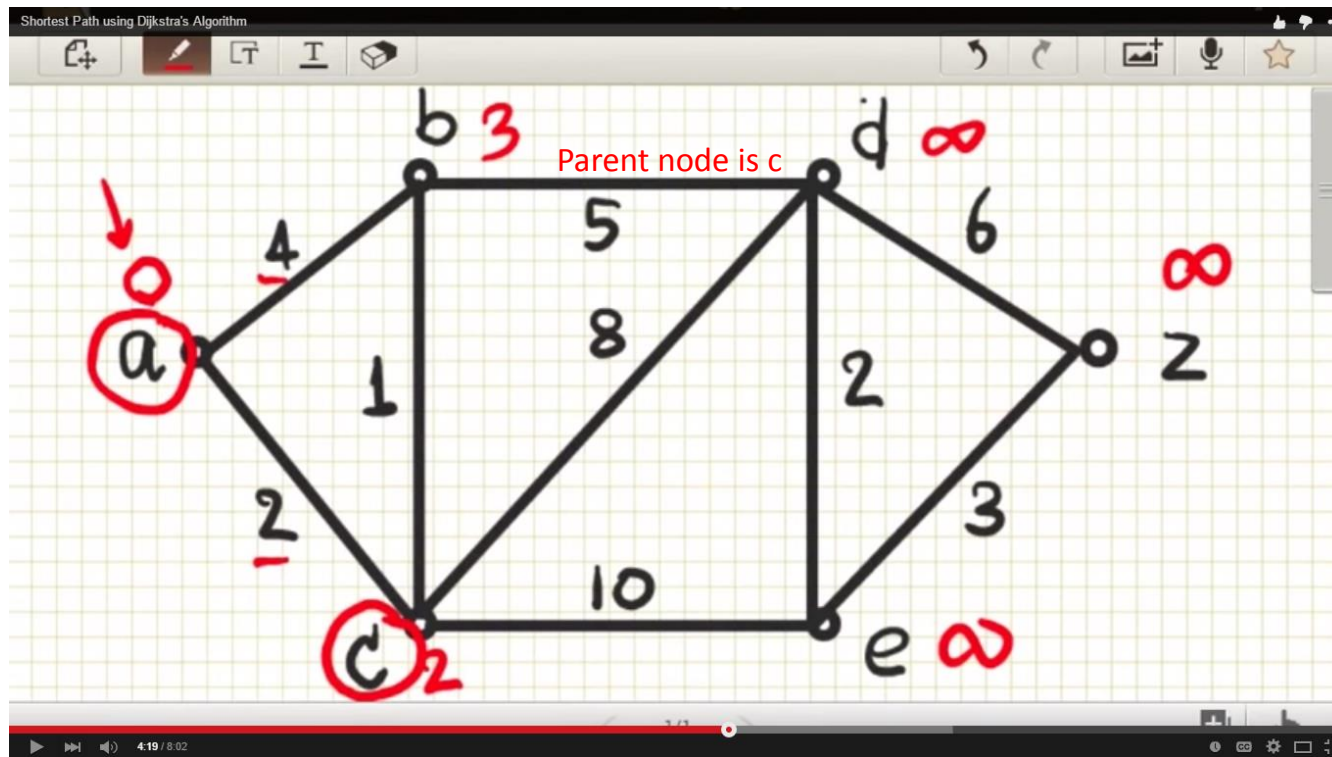
Nodes that are circled are
Said to be visited node
And are never visited once
The distances of all
Its neighbours are updated.



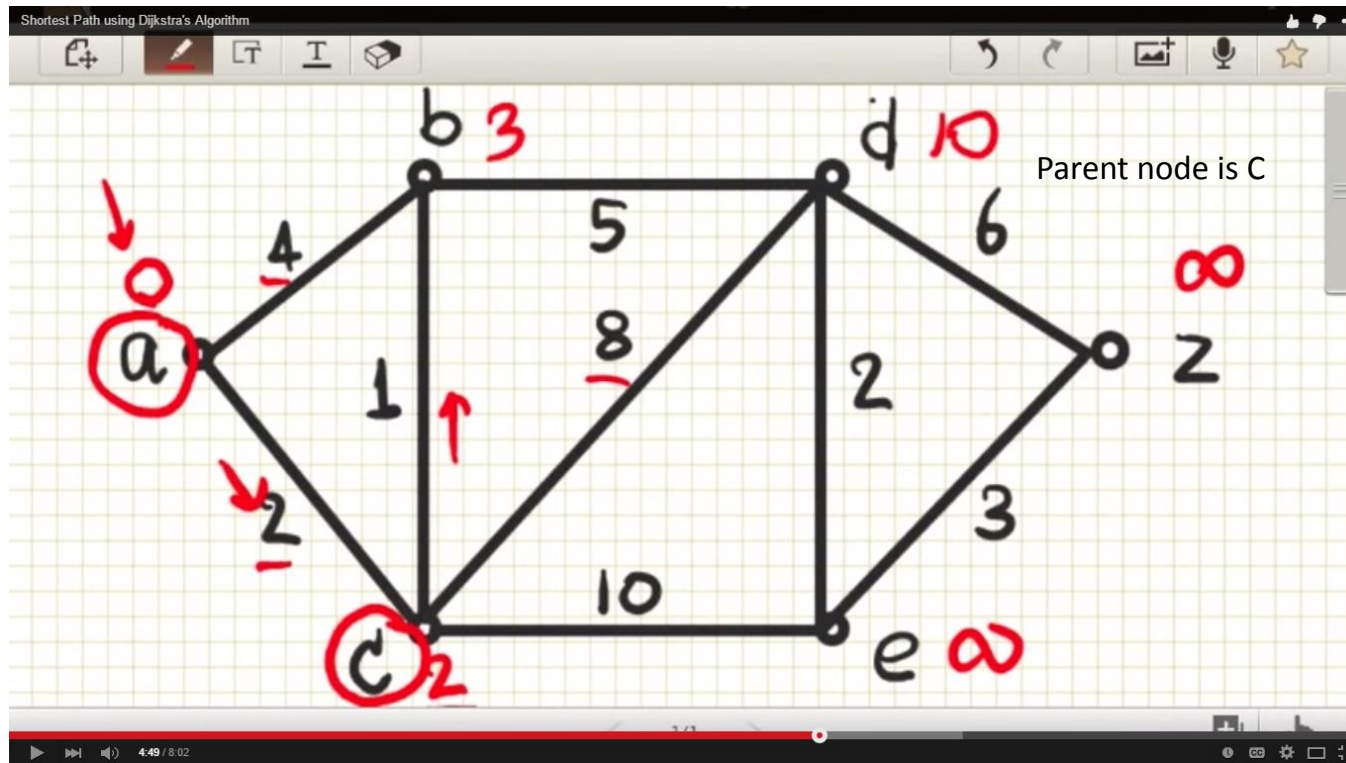
Here The distances of node B and Node C are updated As their distances (infinite) Was more than the distance from the current Node A.

Node A is marked as visited

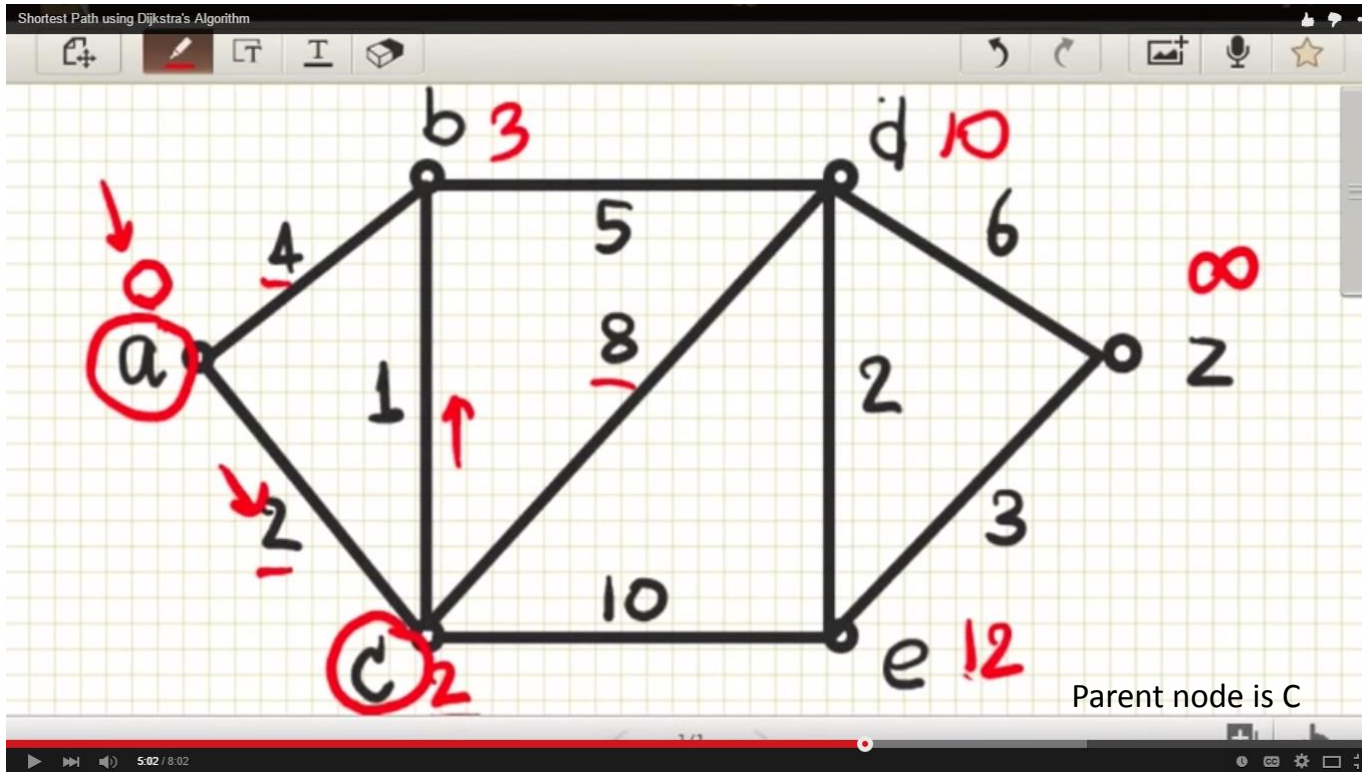




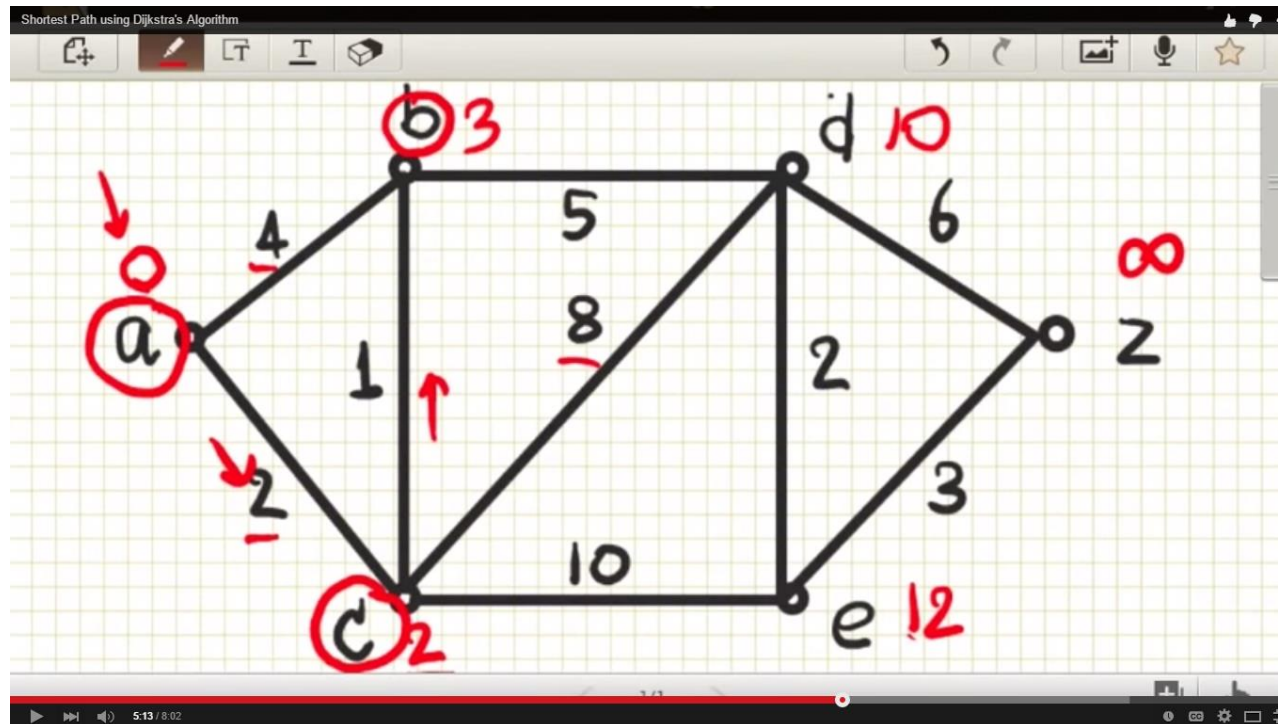
Distance of node
B is updated



Distance of node D
Is updated here

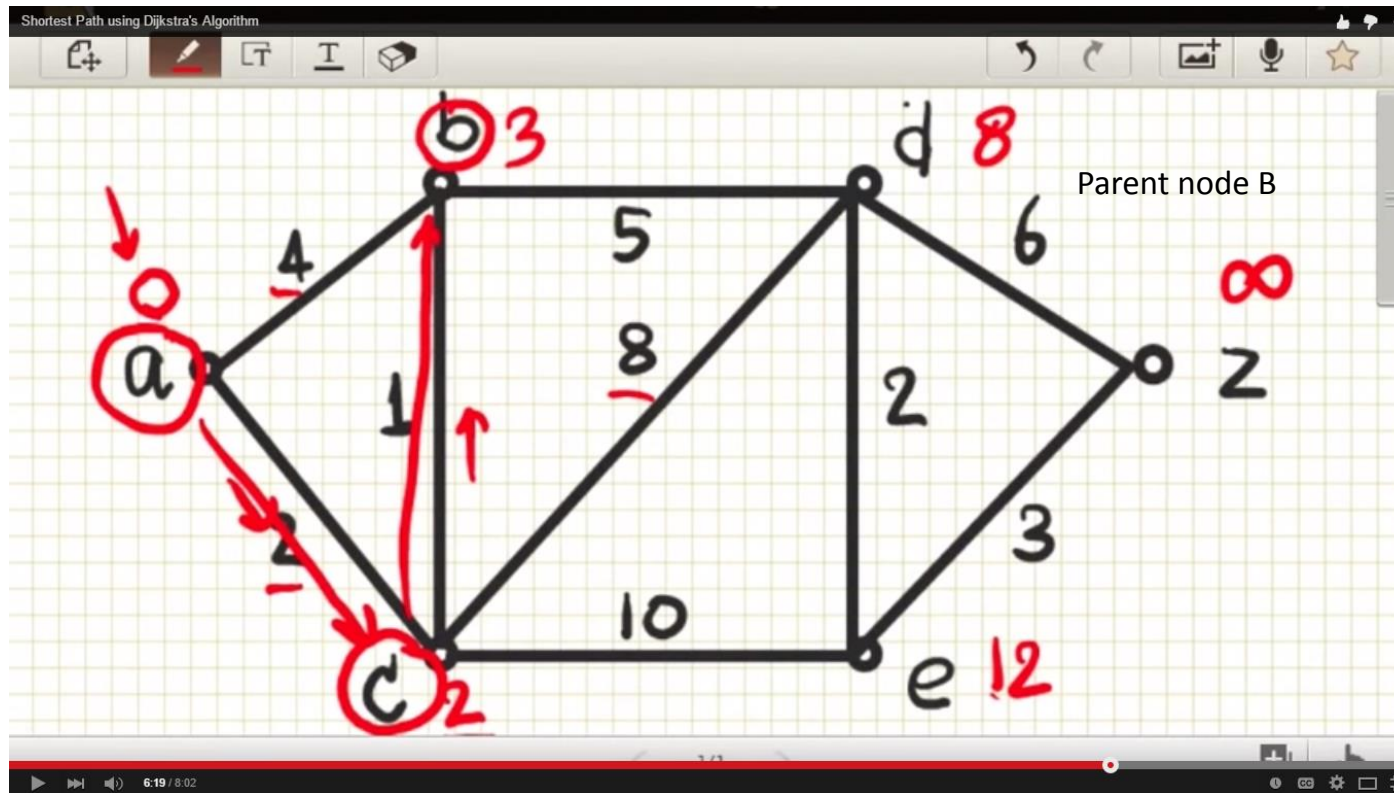


Distance of Node E is Updated here.

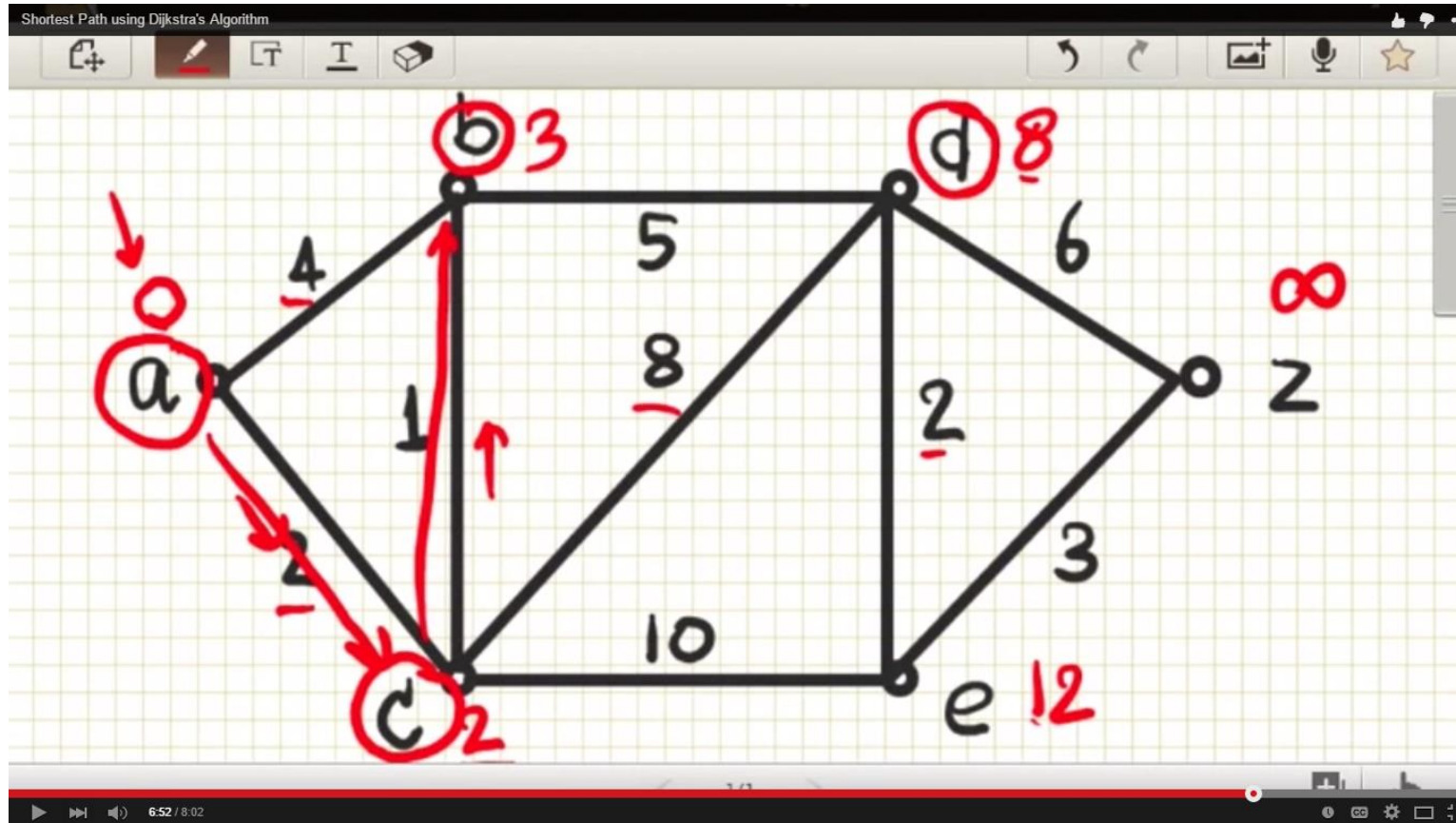


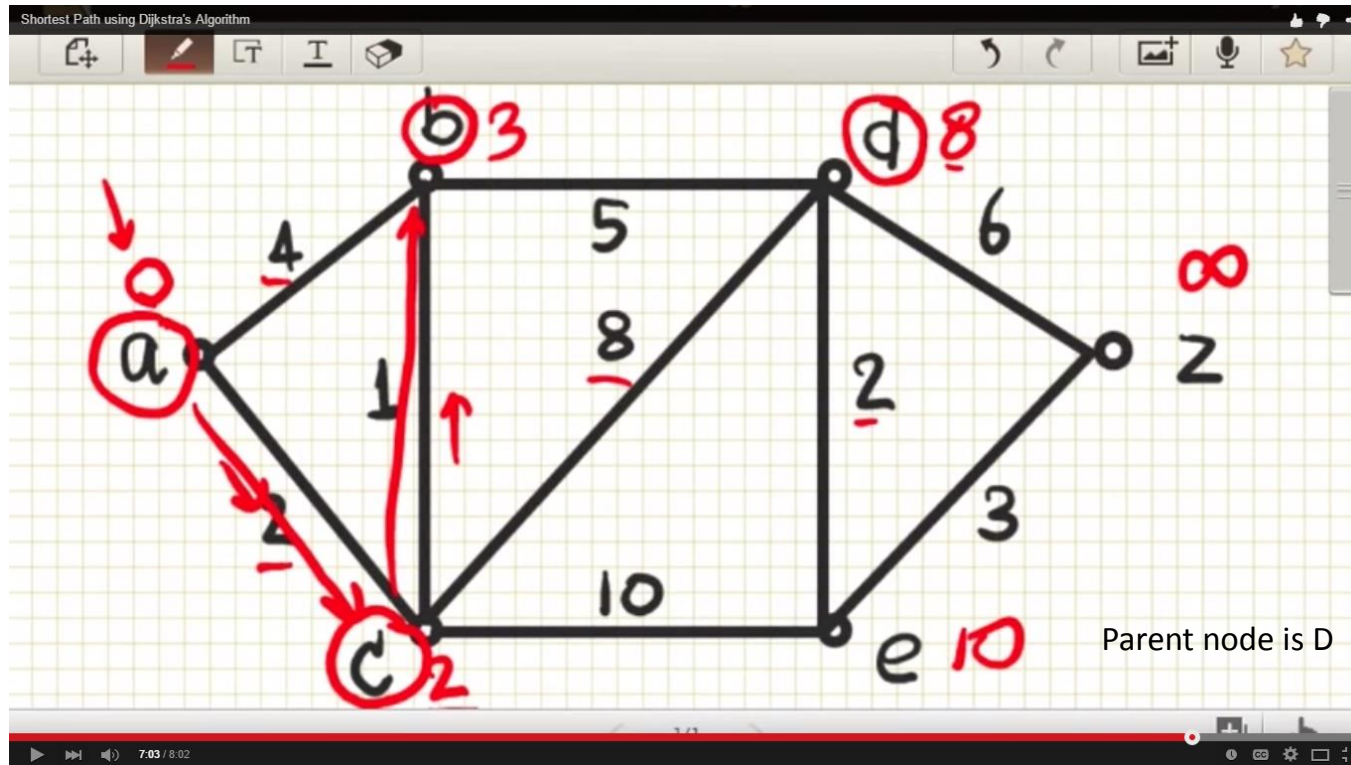
As distances of all the neighbouring nodes Of node C are updated ,it is marked visited.

Node B is then selected as The current node.

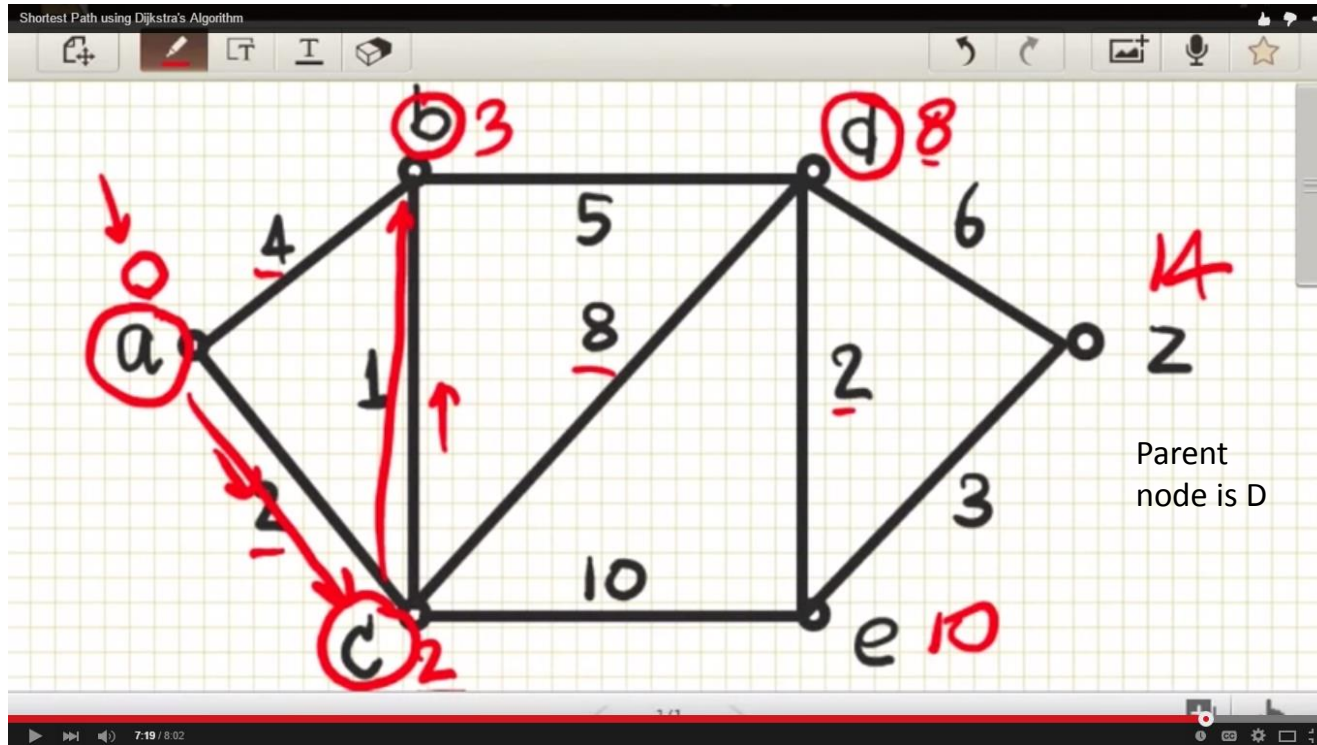


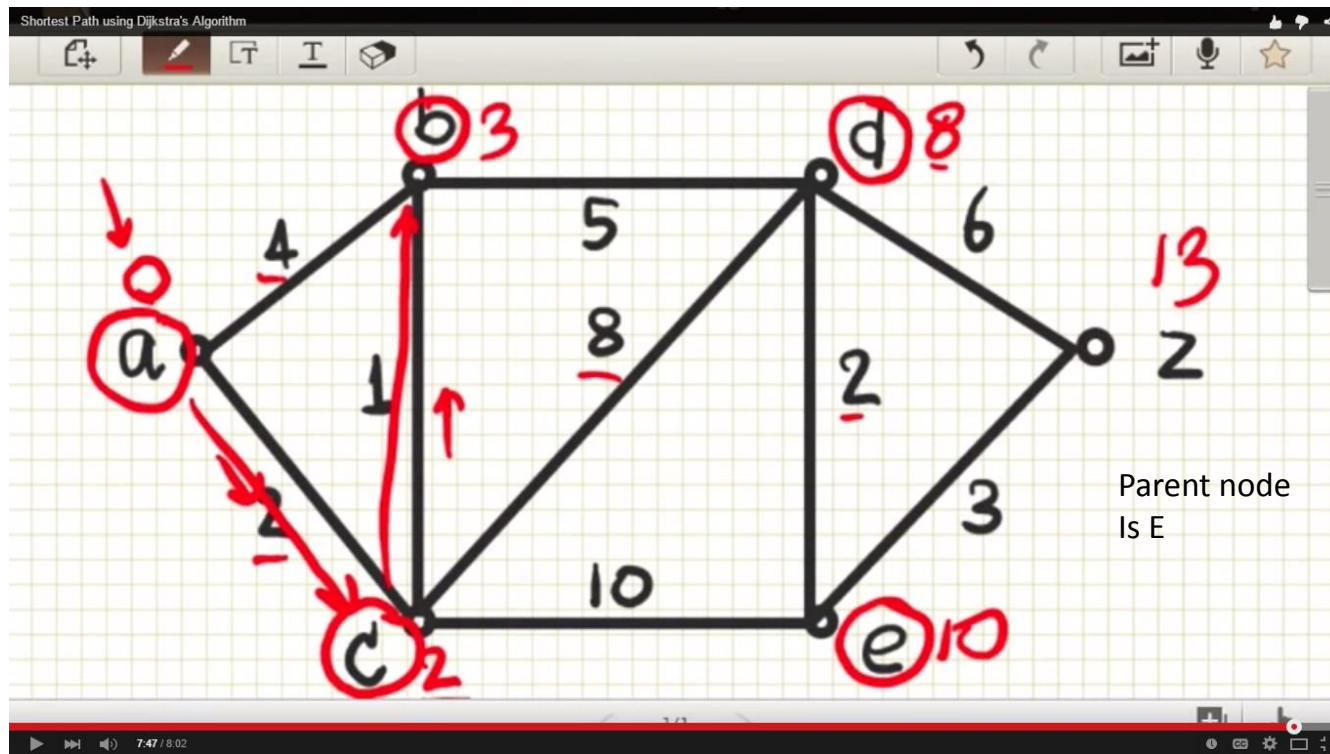
Node D updated.

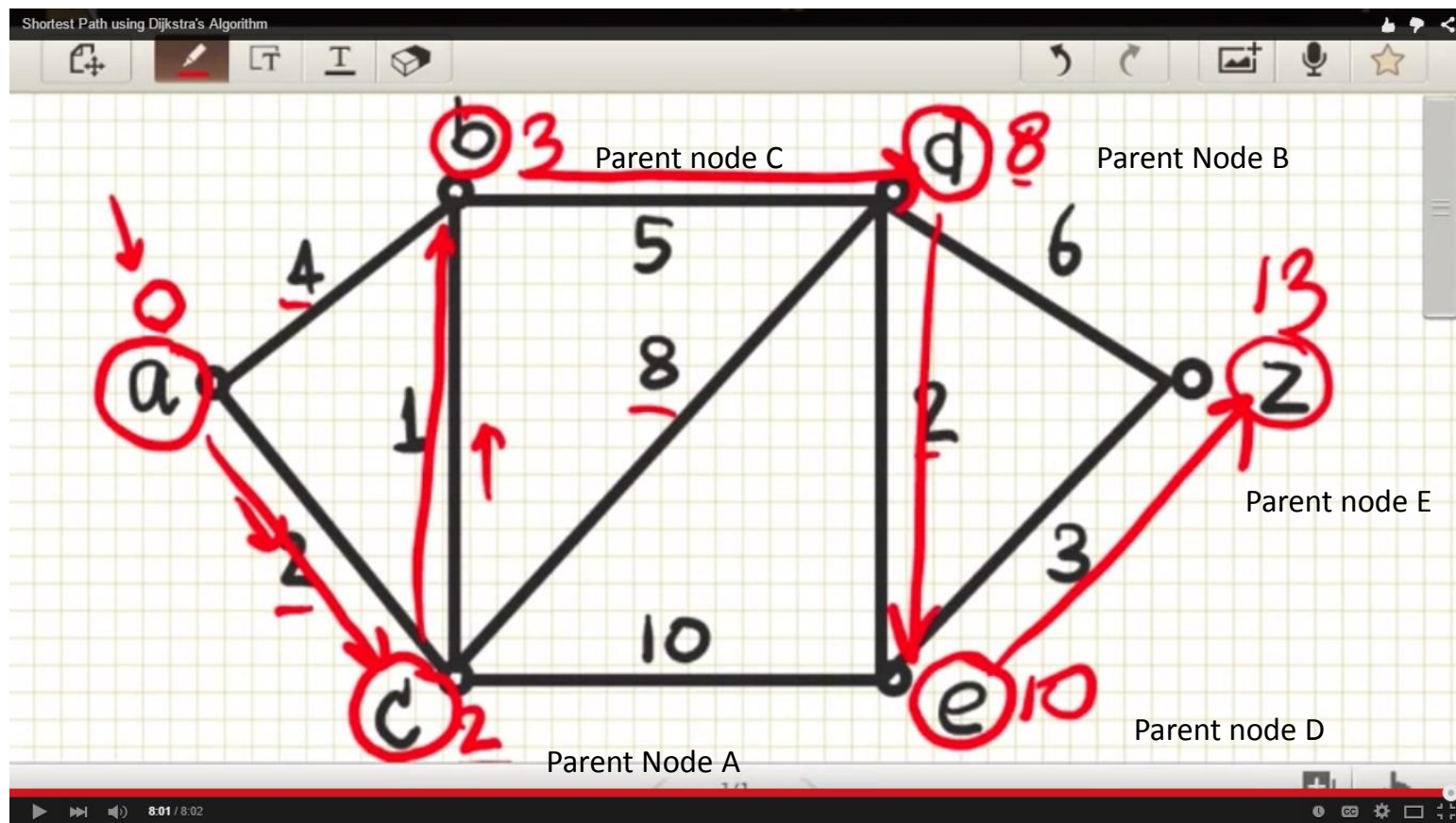




Node B is marked
Visited and node
D is the current node







Need to create 4 matrices,

1. Matrix to store which whether a node can be visited or not.
2. Matrix to store whether a node which can be visited(indicated by 1 in the previous matrix) Is actually visited or not.
3. Matrix to store the distances of the nodes.
4. Matrix to store the base address of a node(parent node address).

```
1 - scr = serial('COM4', 'BaudRate', 9600);  
2 - fopen(scr);  
3 - pause(2);  
4 - fwrite(scr, 's');  
5 - pause(1*30/35);  
6 - fwrite(scr, 'a');  
7 - fclose(scr);|
```

Contacts:

Mayank Garg: 9416244525

Prateek Kumar: 9931417831