

Video Processing

We have done a lot of cool things with processing up to this point. Most of it has been with shapes and manipulating them. Thats fine, but those shapes really aren't dynamic. Video would be an awesome material to use! So lets do it!!!



PROCESSING

SKILL REQUIREMENTS

DIFFICULTY 1-5

ELECTRICAL
PROTOTYPING



ROBOTICS



SOLDERING



PROGRAMMING



DIY



MATERIALS LIST

- Computer
- Processing
- WebCam
- QuickTime 7 or higher

STEP 1: What is a library?

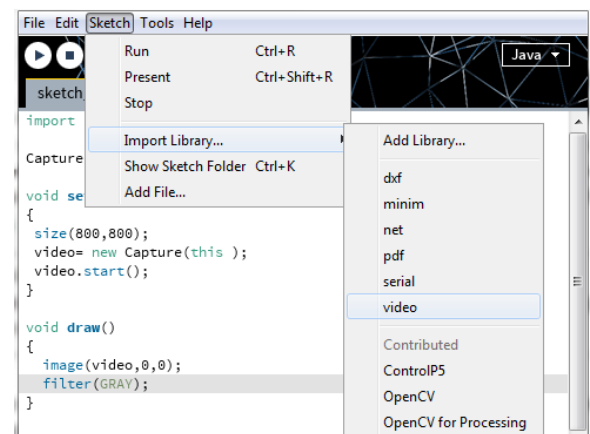
When you get deeper into writing code you start to find that you have more specific needs in your endeavors, or you want to connect your project to other things like an Arduino, a webcam, or even a Microsoft Kinect. For using these peripherals you could write 1000's of lines of code, or you could use what is called a library. A library is a large collection of pre-written code that in many circumstances boils hundreds of lines of code down to one function for you. The libraries are written by anyone and you can find them anywhere on the web. The hardest part of using a library is getting it added to your sketch and figuring out how each one works. For your first example of using a library we will look at the video library in Processing!

STEP 2: Preflight Check

Before working with the video library it is a good idea to make sure that your webcam works on its own. Open your favorite video editing/ recording software and make sure that you can view yourself using the camera that you intend on using. This may be a built in camera or a USB, it doesn't matter! Also, make sure you are up to date with QuickTime, like it or not this library makes use of it!

STEP 3: Installing a Library

We are using a pre-installed library. If you are using a newer version of Processing you can add a library if you go to **Sketch>Import a library...>Add a Library**. For the sake of simplicity the video library comes pre-installed with Processing. To add the library to our sketch we go to **Sketch>Import a Library...>Video**. You should see the `import processing.video*;` line at the top of your sketch.



STEP 4: Initiating the Library Object

Just like an image or a font we have to initiate the video as an object. We do that using Capture and naming it video (you can name it anything you would like).

STEP 5: Setup

We can now create some `setup()` code for our camera application. In `setup()` we are going to do a few things. We are first going to assign video an “address” or the connection between the application (`this`) and the camera. We can then start the video by calling out the capture library object and a period, then the library function of `start()`.

STEP 6: Draw...a Video?

Drawing the video is super simple at this point. We use the `image()` function. We pass the video object data to the image and set the x,y. We have to also add an event function to get the video working. Under your `draw()` (outside of the `{ }`) create a `captureEvent()`, we are going to also pass it a capture object ‘c’. The only thing inside of the capture event is going to be a `read()` function for the capture object which just tells Processing to read a frame of video when video is being captured.

```
import processing.video.*;

Capture video;

void setup()
{
  size(800,800);
  video= new Capture(this );
  video.start();
}

void draw()
{
  image(video,0,0);
  filter(GRAY);
}

void captureEvent(Capture c)
{
  c.read();
}
```

STEP 7: Video Processing

Before you start going crazy with manipulating the video hit play. It will take a little longer than usual to load due to the use of video. Say Cheese, you are now on camera! Lets do something with the video now! We can add filters to the video just like we did with images using the `filter()` function. As a reminder here are a few of the filters you can use:

`GRAY`- Black and White filter

`POSTERIZE`- Reduce image to a set number of colors

`THRESHOLD`- High contrast black and white

`INVERT`- Camera film style inversion

`BLUR`- Just as it says...Blur!

`DILATE`- High light brightness

`ERODE`- High darness contrast

STEP 7: Going Further!

Adding filters is one thing, but doing some crazy things with video that you have never thought of is the strength of processing video on a computer and in Processing. Check out the example code for the library. If you go to [file>Examples... >Libraries>Video](#) you will find an awesome collection of video based sketches. Take one of those examples and make it your own, add code that you know already, try changing numbers and parameters within the example code. Maybe even combine two examples to make something cool. Ready, set, hack!!!

STEP 8: Share your work!

If you would like to share your work...which you do! You can sign up at openprocessing.org. Open Processing allows you to share your code, view the drawing online, as well as allows others to fork or borrow your code to manipulate and change it on their own while still keeping your sketch intact for others to check out.

OpenProcessing

TAKING IT FURTHER

- try using math and variables to make interaction more intricate
- make a snowman run around in your holiday image