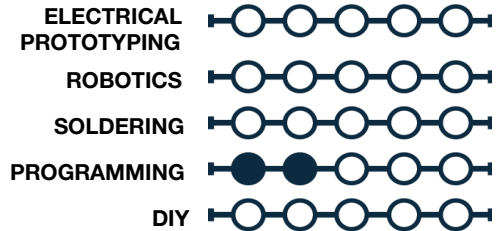


Image Processing

Did you know that you can do image processing and editing in Processing. What makes it even better is that you can apply what you have learned thus far to editing those images and make them interactive.



PROCESSING
SKILL REQUIREMENTS
DIFFICULTY 1-5



MATERIALS LIST

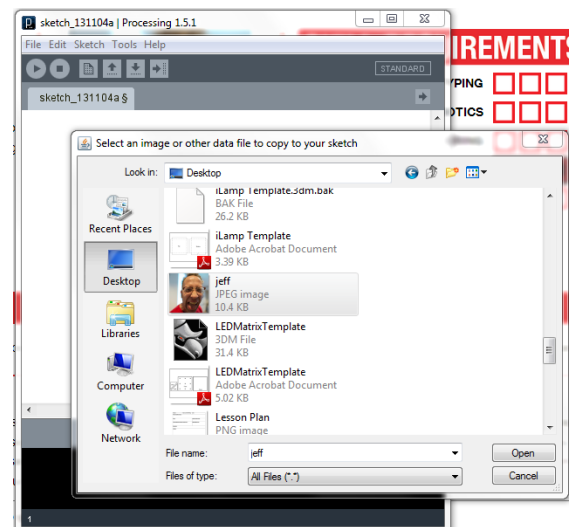
- Computer
- Processing
- Graph Paper
- Colored Pencils
- Super fine tip sharpie

STEP 1: Find an Image

Any image as long as it is .JPG or .BMP will do! If you want to find something online go ahead and Google image search, just take note of the size in pixels and the image format. You need to save the image to somewhere where you know where it is! **Note:** When you save it make sure you name it something **simple** and concise! For an example we are going to use a picture from our Sparkfun flickr feed.

STEP 2: Adding an Image to your Sketch

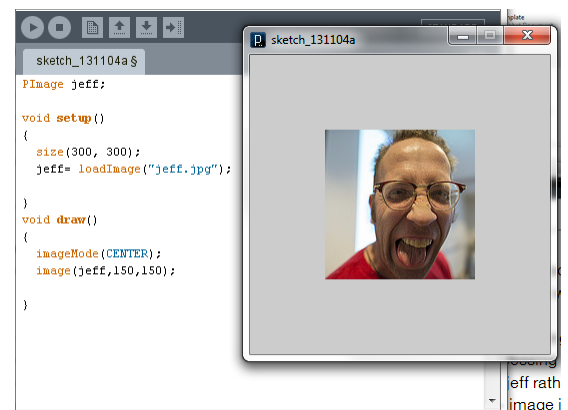
For us to add an image to a Processing sketch we have to do some leg work first. We need to add the image file to our sketch. Any file we add goes into a folder called data in our sketches folder. In Processing go to the **Sketch** drop down menu and select the **add files...** option. We are going to add an image of Jeff, the cool guy at the top of the page! Go and find an image you want to add to your sketch, select it, and click on Open. **Note the file name and type!**



STEP 3: Advanced Data Types

An image can be classified as an advanced data type or an “object”. We have set it up so that Processing can find it by adding it to our data folder, but now we are going to start to bring it into our sketch. In the code window we are going to start typing some functions that are specific to advanced data types and object oriented programming.

We are first going to create a **PImage** object called jeff before the `setup()` code. This is like creating a name tag for someone, we now need to go and stick that name tag on something. In our `setup()` code we tell processing that `jeff= loadImage("jeff.jpg");` In other words we stick the name tag of jeff on an image file and can just use jeff rather than a long file name. We are now ready to draw the image of jeff in Processing. I also knew ahead of time that the image is 250x250 pixels so I am planning on giving it some border space.



STEP 4: Treat it like a Rectangle

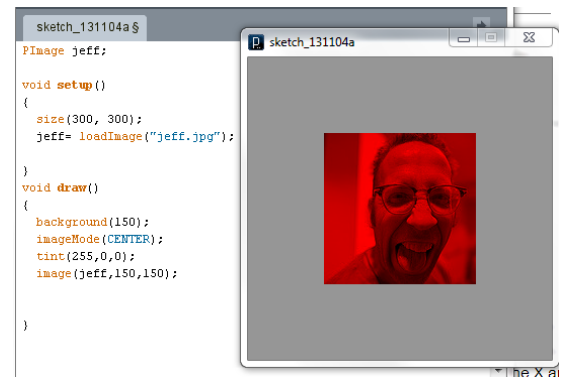
In our `draw()` code we can now use `jeff` in an image function. To make it easier to place the image I used the `imageMode(CENTER)`; function which changes the origin of the image to the center, you can also do this with the `rect()`; function as well for future reference. I can then use the `image(jeff, X, Y)`; function to place an image into the sketch.

STEP 5: Modify an Image

Once you get the image function to produce an image for you, we can play around with it for a bit. First of all, an image functions just like a rectangle so we can pass it two more numbers other than X and Y, we can also set a width and height as well. In your image function add two more numbers after the X and Y and those are going to be the height and width. The complete syntax for image is as follows: `image(file, X, Y, W, H)`; Use `mouseX` and `mouseY` in your image function and see what happens!

STEP 6: Tint or not to Tint?

Cool, now lets start changing this image with the `tint()` function. The syntax for tint is as follows `tint(R, G, B, A)`; Where RGB is the red, green and blue values of the color you want to tint to. A is the Alpha or transparency value of the tint, a note, you give a `fill()` function an Alpha value as well! Just like any other modifier. Check out the difference when we add a red tint modifier to our image of Jeff. Dare I mention using variables with tint?



STEP 7: Filters!

The last and probably most creatively broad function is the `filter(TYPE)`; function. This function is a little different than the other modifier functions. This is placed after the `image()` function rather than before it! Filter has a number of different types See below for the types and example of what they look like when applied to an image.



Now its your turn, take you image and manipulate it using the tint and filter functions, once you feel comfortable try using variables and math to make your image change over time.

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.



TAKING IT FURTHER

- Use both the `tint()` and `filter()` function together to make some pretty cool images!
- Use multiple images with different filters!
- play around with `text("text", x, y)`;