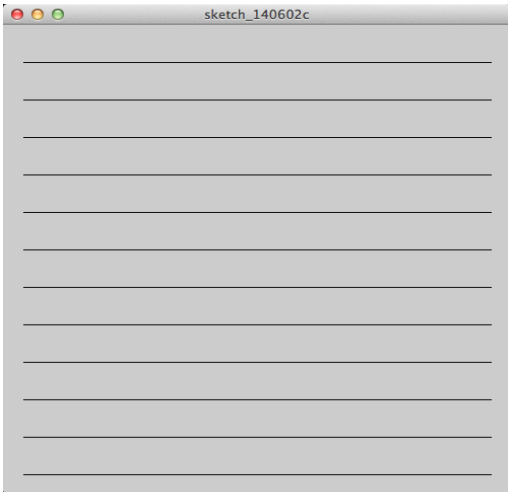
5 - Images in Processing

Overview: We are going to take photo of ourselves and use images from the internet to learn about importing images into processing and make alterations adding layers and fades.

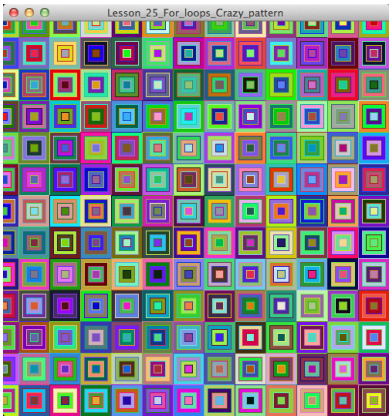
STEP 1: RECAP SESSION 4 - PATTERNS - FOR ( ) FUNCTION

We are able to use iteration to create the same objective but with using only a few lines of code. At first it looks complicated, but we will explain it step by step. Type in the code below and run the program.




```
void setup() {
  size(500, 500);
}

void draw()
{

   for (int i = 0; i < 500; i = i+40) {
    line(20, i+40, 480, i+40);
  }
}
```

The final part of the last session was to allow you the chance to experiment with adding multiple shapes and controlling each one with random colours. You can add as many as you want, but remember the shapes at the top of your code will be in the background and each line of code will place the shape in front.




```
void setup() {
  size(500, 500);
  frameRate(10);
  rectMode(CENTER);
}
void draw()
{
  for (int i = 0; i < 500; i = i+40) {
   for (int j = 0; j < 500; j = j+40) {
     fill(random(0,255),random(0,255),random(0,255));
     stroke(random(0,255),random(0,255),random(0,255));
     rect(i, j, 40, 40);
     fill(random(0,255),random(0,255),random(0,255));
     stroke(random(0,255),random(0,255),random(0,255));
     rect(i, j, 30, 30);
  }
 }
}
```

STEP 2: TAKE IMAGES OF YOURSELF

In Processing we can import images and alter them the same we you can in many photo editing tools, such as Adobe Photoshop.

The difference here is that we get to choose the filters and code them ourselves, giving us maxium control over the photo manipulation tools.

First lets use a photo of yourselves. You can either use a photo you have already taken or you can take a one now.

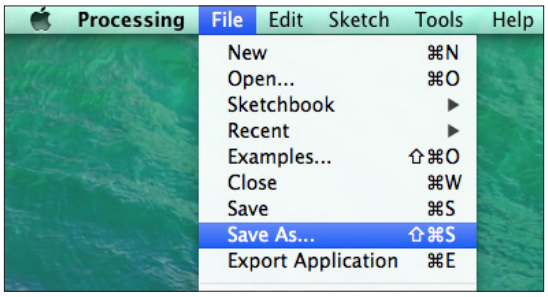
On an Apple Mac you can use the Photo Booth software. On a windows computer you can use many different software programs, one such software could be ManyCam utility (http://download.manycam.com/?os=win).

Save your photo to a place you will remember.

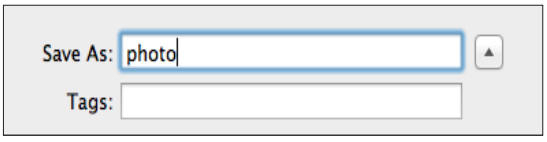
Next lets select several images from the internet. You can either use Google image search or your own images.

For us to manipulate images in processing we will first need create a processing sketch and move our images into the same folder.

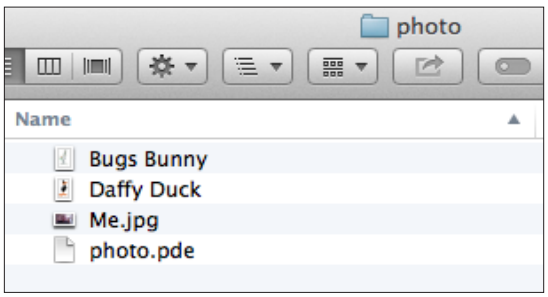
You will need to open up Processing and before we write any code we will create a new file and save it automatically.



You should name you Processing Sketch "photo" and then press Save.



Move over any images you want to use in your Processing programme into your photo folder. Here I have 4 files, 3 images and a processing file .pde. It should look like the image on the right.

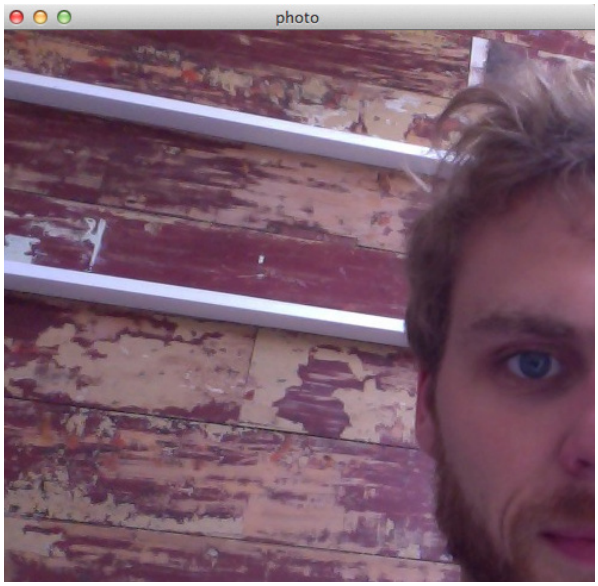


STEP 3: LOADING IMAGES INTO PROCESSING

Now it is time to add your images into your Processing Sketch. Processing can display four different types of images, these are **.gif**, **.jpg**, **.png** and **.tga**. So far your images will be .jpg and .png.

First we will need to declare a variable using `PImage()`, we can call this anything we want, but in this case we are going to call it "photo".

```
PImage photo:

void setup() {
  size(500, 500);
  photo = loadImage("Me.jpg");
}

void draw() {
  image(photo, 0, 0);
}
```



Use the `loadImage()` function in the **`setup()`** function, specifying which images we want to use, in this case we have used an image of you. Remember that your image has to be in the same folder as your Processing sketch.

The image() function draws the image to the specific place we choose and we can also scale in the images using the next two variables.

```
image (PImage, x-coordinate, y-coordinate, width of image, hight of image);
```

note > If you find your image is cropped and you want to see the full image you will need to change the size of your window to the same as image.

STEP 4: TINTING

The next this we will do is control the layers between two images. We can make one image a background and then the second image will be half the opacity, allowing you to see through the image, creating a merged affect.

Lets add a secod image into our code, we can use the same `PImage` function. Mine is called cat, but you should use the name of your file your going to use.

```
PImage cat:
```

Next you will add the `loadImage()` function for your seond image into your **`setup()`**. You will also change the size of your project to the same as your first image.

```
size(1080, 720);
photo = loadImage("Cat.jpg");
```

In the **`draw()`** function you will add the photo of yourself as a background image and add a Tint.

```
tint( a, b);
```

a = 255 is the colour

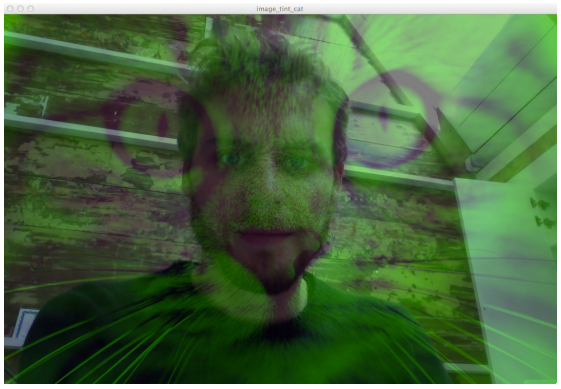
b = change the number below 255 to add opacit, the lower the number the more the background with show through.

```
PImage cat:
PImage photo:

void setup() {
  size(1080, 720);
  Me = loadImage("Me.jpg");
  Cat = loadImage("Cat.jpg");

}

void draw() {
  background(Me);
  tint(255,127);
  image(Cat,0,0);
}
```

? Can you think of two images which would look amazing merged together? You could use more than two image!

Next we will add more options to our tint, adding different colour options using Red, Green and Blue (RGB) details.

```
tint( aR, aB, aG, b);
```



a R= RED choose between 0 -255
aB = GREEN choose between 0 -255
aG = BLUE choose between 0 -255

b = change the number below 255 to add opacity, the lower the number the more the background with show through.

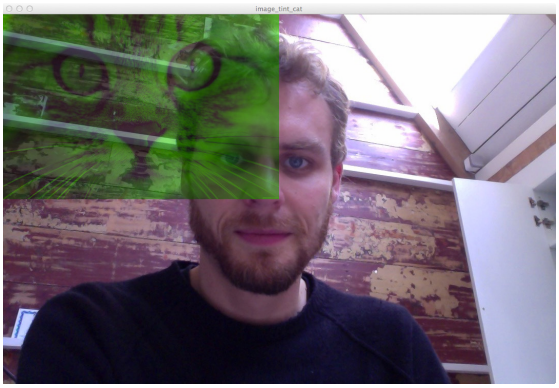
```
tint( 0, 0, 255, 112);
```

In this example the colour is green with a transparency.

? Can you alter the colours in your layers?

STEP 5: SCALE

You are able to rescale both the hight and width of your images in processing,



For example, to change the size of the image of the person:

```
imagename.resize( a, b);
```

a = Width of the image
b = hight of the image

```
void draw() {
  image(photo, 0, 0);
  background(Me);
  tint(255,127);
  Cat.resize(0,360);
  image(Cat,0,0);
}
```

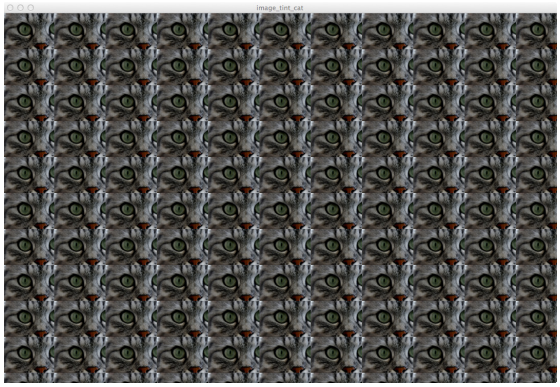
note > Now you can add many more images together and see what crazy patterns you create. Don't forget to experiment with different fades and scaling your images!

STEP 6: EXTRA - IMAGES AND PATTERNS

In previous session we have learnt about using the `for()` function. For this extra challenge, you will demonstrate how to use both images and creating patterns.

In our **`draw()`** function all we need to do is add our two `for()` function, excatly as we did in our previous session. This time when we call up the x and y of our image position, we use the letters we have added in our `for()` function.

```
PImage cat:

void setup() {
  size(1080, 720);
  Me = loadImage("Me.jpg");
}

void draw() {
   for (int i = 0; i < 500; i = i+540) {
      for (int j = 0; j < 500; j = j+360) {
        Cat.resize(0,360);
        image(Cat,i,j);
      }
   }
}
```



? Can you create other patterns using your images, you can experiment with scale, transparency, layers, colour and creating patterns, incorporate all these eliments into a final image.

DON'T FORGET TO SAVE YOUR WORK FROM THIS SESSION!