# 258 Labs

## 01 Introduction to Processing

**Part 1**

For this part you will require the BalloonExample folder available in the 01 Examples.zip file on Moodle.

Steps:

1. Extract BalloonExample to your 258 work folder or desktop and ensure that it works properly, i.e. it shows an animation of a balloon falling to the ground.
2. The colours for the balloon are set in the setup() method of the program. Change them by using the Color Selector tool to select new colours.
3. The balloon always starts at the same x and y position. Use random() (<http://www.processing.org/reference/random_.html>) to initialize the balloon to a random x position between 0 and width, and a random y position between 0 and height/2.
4. Having separate x and y variables of type float to store the location of the balloon on screen is not very efficient. Read the online description of the PVector class (<http://processing.org/reference/PVector.html>). Modify your program so that it uses a single PVector position instead of x and y. (Note: you should completely remove x and y from your program.)
5. Having one balloon is not very exciting. Modify your program so that it can animate multiple falling balloons at once. You should define a constant NUM\_BALLOONS to specify how many balloons to display. Each balloon should start at a random (different) location and fall to the ground. You will need to use **arrays** and **loops** to solve this problem. The easiest way is to define an array of PVectors for the positions of the balloons, and then use a loop (i) in the setup method to initialize each balloon’s position, and (ii) in the draw method to draw and move each balloon.
6. Modify your program so that each balloon also has a random colour. You will need to define an array of colors for this:  
   color[] fillColours = new color[ NUM\_BALLOONS ];  
   and then a random colour can be assigned with astatement like this:  
   fillColours[index] = color(random(255),random(255),   
    random(255));

You should demonstrate your program with 10 balloons of different colours falling in order to get marks for this part.

**Part 2**

1. Investigate the PImage class (<http://processing.org/reference/PImage.html>). You can add images to your sketch by dragging them onto the Processing window and then using the loadImage() function to read the image into memory.
2. Create a new blank sketch. Learn about images by loading an image into your sketch inside the setup() function and then drawing it at (0,0) inside the draw() function. You will need to declare the image as a global variable so that it can be accessed in both functions.
3. Learn about the imageMode() function (<http://processing.org/reference/imageMode_.html>) by testing the effects of different image modes when you draw the image at (0,0).
4. After loading the image, resize the image using image.resize(,) so that its width is 10% the width of the sketch. Check the PImage page for details on how to resize.
5. Write a function tile() that will tile the sketch with copies of the image. The function should be called with the number of copies of the image that you want, e.g. tile(image,10,10) will tile the sketch with 10 images across and 10 down but tile(someImage, 5,3) will only tile five images across and five down.

Demonstrate your tiling function to the tutor by running your program at least twice with different images and different values for the number of images across and down, in order to get marks for this part.