Assignment 1

This assignment is due on February 4^{th}, late submissions will not be accepted. **Please include all references, and the names of all students in your study group. Remember all programs should be submitted as an IPython (.ipnb) as well as the pdf file. Your assignment should be a single zip file containing EVERYTHING needed to run your code.**

1. Use the **Scipy** module to load the ultrasound dataset "Ultrasound.mat", the variable that you are after is called "B". **(Score: 20 points)**
   (a) Reshape the data set as a $601 \times 750 \times 27$ data set
   (b) Use subplot to display frames 10 to 13; remember to use the grayscale colormap to display the images and properly threshold the ultrasound B-mode images (minimum value = -40, maximum value = 0).
   (c) Use the **skimage** module to resize all the frames as 128 by 128.
   (d) Resize just the frame 10 to 1024×1024, 512×512, 128×128, 32×32; what effects do you observe when the image get smaller and smaller
   (e) Write a python function for displaying the resized data set as a Montage, i.e., a simple image containing all frames with a spacing of 20 pixels in both directions between the images.

2. Finding the maximum, minimum values in an image, use the Hubble eXtreme Deep Field scikit-image for this problem (remember to convert from RGB to grayscale): **(Score: 20 points)**
   (a) Write a python function for finding the maximum and minimum values each column of an images,
   (b) Write a function for scaling an image between any two values, please store the original and scaled image as a tuple (**a list containing both images**), and provide comments on how your algorithm works.
   (c) Save the output file as a MATLAB binary, and an compressed **numpy** binary file,

3. Use the RGB image coffee available in Python for the following exercise: **(Score: 20 points)**
   (a) Use subplot to display each of the three channels.
   (b) Write a simple function for converting RGB images to grayscale (no loops), you cannot use the skimage rgb2gray function
   (c) Do a simple comparison to see how your grayscale image compare to that produce using the rgb2gray function