Assignment 1: Building a Better Contact Sheet

In the lectures for this week you were shown how to make a contact sheet for digital photographers, and how you can take one image and create nine different variants based on the brightness of that image. In this assignment you are going to change the colors of the image, creating variations based on a single photo. There are many complex ways to change a photograph using variations, such as changing a black and white image to either "cool" variants, which have light purple and blues in them, or "warm" variants, which have touches of yellow and may look sepia toned. In this assignment, you'll be just changing the image one color channel at a time

Your assignment is to learn how to take the stub code provided in the lecture (cleaned up below), and generate the following output image:

Code & Output

In [16]:

```
import PIL
from PIL import Image, ImageDraw, ImageFont
from PIL import ImageEnhance
import numpy as np

# read image and convert to RGB
image=Image.open("readonly/msi_recruitment.gif")
image=image.convert('RGB')

# Split into 3 channels
r, g, b = image.split()
fracs = [0.1, 0.5, 0.9]
images = []
j = 0
for i in range(9):
    if i < 3:
        r1 = r.point(lambda x: x * fracs[j])
        g1, b1 = g, b
    elif i < 6:
        g1 = g.point(lambda x: x * fracs[j])
        r1, b1 = r, b
    else:
        b1 = b.point(lambda x: x * fracs[j])
        r1, g1 = r, g
    j = (j + 1) % 3
    # Recombine back to RGB image
    result = Image.merge('RGB', (r1, g1, b1))
    images.append(result)

# create a contact sheet from different brightnesses
first_image = images[0]
gap = 60
contact_sheet = PIL.Image.new(first_image.mode, (first_image.width*3,first_image.height*
3+gap*3))
x, y = 0, 0
draw = ImageDraw.Draw(contact_sheet)
font = ImageFont.truetype(r'readonly/fanwood-webfont.ttf', 75)

j = 0
for img in images:
    # Lets paste the current image into the contact sheet
    contact_sheet.paste(img, (x, y) )
    text = 'channel {} intensity {}'.format(j // 3, fracs[j % 3])
    imga = np.array(img)
    color = int(np.mean(imga[...,0])), int(np.mean(imga[...,1])), int(np.mean(imga[...,2
]))
    # drawing text size
    draw.text((x, y+img.size[1]), text, font = font, align ="left", fill=color)

    # Now we update our X position. If it is going to be the width of the image, then we
set it to 0
    # and update Y as well to point to the next "line" of the contact sheet.
    if x+first_image.width == contact_sheet.width:
        x=0
        y=y+first_image.height+gap
    else:
        x=x+first_image.width
    j += 1

# resize and display the contact sheet
contact_sheet = contact_sheet.resize((int(contact_sheet.width/2),int(contact_sheet.heigh
```

```
t/2) ))
display(contact_sheet)
```

In [ ]: