

Today we'll have a hands-on lab to familiarizing you with writing picture filters, so **pairs of students** should get a laptop from the cart. (NOTE: laptops are numbered and plugged-in in back -- plug-in and put away in their proper spot)

Download to the Desktop and extract the files needed for the lab from:

<http://www.cs.uni.edu/~fienup/cs1120s15/sessions/s07/lec7.zip>

**Part A:** Launch JES, then in the Interactions area type:

```
>>> pickAColor()
```

This should pop-up the "Choose a color" window. It is initially in "Swatches" mode showing small squares of a bunch of different colors. Click the RGB tab to change the view. In RGB mode we'll be interested in the Red, Green and Blue sliders which we can adjust to display any of the  $256 * 256 * 256 = 16,777,216$  available colors in the objects at bottom of the box.

a) Play with the sliders and see what RGB color values you think produce the following colors:

Color	R value	G value	B value
brown			
"grass green"			
"sky blue"			

b) Once you have found RGB values you like for the above colors, you can use the `makeColor` function to create customized colors of your choosing. The `makeColor` function expects three numeric parameters (the R, G, and B values) of the color. For example, the following code shows a UNI Gold canvas with a UNI Purple square in the middle. (You can load and run the program `uniGoldAndPurple.py`)

```
>>> uniGold = makeColor(255, 204, 0)
>>> uniPurple = makeColor(75, 17, 111)
>>> canvas = makeEmptyPicture(400, 400, uniGold)
>>> addRectFilled(canvas, 50, 50, 300, 300, uniPurple)
>>> repaint(canvas)
```

Create a similar program to create a sky blue canvas with bottom half filled with green grass and a brown log (rectangle) laying on the grass.

c) In the Interactions area type the following statements:

```
>>> newColor = makeColor(500, 100, 300)
>>> print newColor
```

What does the print statement display?

Explain `newColor`'s resulting RGB values?

d) In the Interactions area, what gets printed when you type the following statements:

```
>>> getColorWrapAround()
>>> setColorWrapAround(true)
>>> getColorWrapAround()
>>> c = makeColor(500, 100, 300)
>>> print c
```

Explain `newColor`'s resulting RGB values?

**After you complete Part A, raise your hand, demonstrate your program (part b), and answers.**

**Part B:** Write a program with a new filter function called `grayThenNegate` which changes a picture to grayscale and then negates it. Use the simple grayscale calculation of setting each pixel's new RGB values all to the average of its original RGB values.

I suggest that you start with a blank program to practice writing a complete program. You can model your program after either `simpleGrayscaleFilter.py` or `negateFilter.py`.

**After you complete Part B, raise your hand and demonstrate your `grayThenNegate` filter program.**