Python: Working with pixels
Reminder: Conditionals

```python
if age < 18:
    showInformation("Sorry, not allowed to vote yet.")
else:
    showInformation("Please select candidate.")
```
Relational Operators

- $a > b$
- $a < b$
- $a == b$ (equality)
- $a <= b$ ("less than or equal")
- $a >= b$ ("greater than or equal")
- $!= $ (inequality)

**Note:** The order of signs matters: $!=$, $<=$, $=>$ do not work!

There should be no spaces inside the relational operator, so "$a <= b$" also does not work (See also Appendix A.4, p362)
Getting the leading zero into the filename:

```python
if n < 10:
    filename = "swatch0" + str(n) + "\n.jpg"
else:
    filename = "swatch" + str(n) + "\n.jpg"
```
Getting the leading zero into the filename:

```python
if n < 10:
    filename = "swatch0" + str(n) + ".jpg"
else:  # here we know n >= 10
    filename = "swatch" + str(n) + ".jpg"
```

What if you need more frames?

```python
if n < 10:
    filename = "swatch00" + str(n) + ".jpg"
else:  # here we know n >= 10
    if n < 100:  # here n is in the range of 10 ... 99
        filename = "swatch0" + str(n) + ".jpg"
    else:  # here we know n >= 100
        filename = "swatch" + str(n) + ".jpg"
```
Examples from lab 11

• Be sure to review these!
• Open examples.py in JES
• Load
• In the command area, try the functions
Getting at the pixels

`getPixel(picture, x, y)`

Gets a single pixel – returns pixel at position `x,y` of `picture`

`getPixels(picture)`

gets *all* the pixels – returns an array containing all the pixels of `picture`
Reminder: Manipulating Pictures

```python
>>> pic1 = makeEmptyPicture(200, 100)
>>> seafoam = makeColor(153, 255, 204)
>>> setAllPixelsToAColor(pic1, seafoam)
>>> addText(pic1, 30, 50, "hello")
>>> show(pic1)

>>> pic2 = makePicture(pickAFile())
>>> show(pic2)
```

Links to small images you can use to test your program:  [eye.jpg](#), [luca.jpg](#)
>>> p = getPixel(picture, 12, 9)
>>> print p
Pixel, red=108 green=86 blue=142
What can we do with a pixel p?

- `getRed(p)`
- `getGreen(p)`
- `getBlue(p)`

- `setRed(p, value)`
- `setGreen(p, value)`
- `setBlue(p, value)`

- Functions that take a pixel (p) as input and return a value between 0 and 255

- Functions that set the value of pixel (p) to a given value between 0 and 255
We can also get, set, and modify Colors

- **getColor(p)** takes a pixel as input and returns a Color object with the color at that pixel.
- **setColor(p, c)** sets the color of pixel (p) as input and a color (c), then sets the pixel to that color.
- We also have functions that can **makeLighter(c)** and **makeDarker(c)** an input color.
- Last time we saw that we can also create colors:
  - **makeColor(r, g, b)** takes red, green, and blue values (in that order) between 0 and 255, and returns a Color object.
  - **pickAColor()** lets you use a color chooser and returns the chosen color.
>>> pixel = getPixel(picture, 12, 9)
>>> print pixel
Pixel, red=108 green=86 blue=142

>>> value = getRed(pixel)
>>> setRed(pixel, value+50)
>>> setGreen(pixel, 0)
>>> setBlue(pixel, getBlue(pixel)/2)
>>> pixel=getPixel(picture,12,9)
>>> print pixel
Pixel, red=108 green=86 blue=142
>>> value = getRed(pixel)
>>> setRed (pixel, value+50)
>>> setGreen(pixel, 0)
>>> setBlue(pixel, getBlue(pixel)/2)
```python
>>> pixel = getPixel(picture, 12, 9)
>>> print pixel
Pixel, red=108 green=86 blue=142

>>> redValue = getRed(pixel)
>>> greenValue = getGreen(pixel)
>>> blueValue = getBlue(pixel)

>>> newColor = makeColor(redValue + 50, 0, getBlue(pixel) / 2)
>>> setColor(pixel, newColor)
```
>>> pixel = getPixel(picture, 12, 9)
>>> print pixel
Pixel, red=108 green=86 blue=142
>>> redValue = getRed(pixel)
>>> greenValue = getGreen(pixel)
>>> blueValue = getBlue(pixel)
>>> newColor = makeColor(redValue+50, 0, getBlue(pixel)/2)
>>> setColor(pixel, newColor)
Repeating an action for all the pixels in a picture

**Example:**

```python
for p in getPixels(picture):
    value = getRed(p)
    setRed(p, value*0.5)
```
Repeating an action for all the pixels in a picture
decreaseRed()

Example:

def decreaseRed(picture):
    for p in getPixels(picture):
        value = getRed(p)
        setRed(p, value*0.5)
More examples:

More examples (link) - you can copy and paste these to save time

• decreaseGreen()
• decreaseBlue()
• clearBlue()
• lighten()
• darken()
• negative()
• grayScale()
“Posterize” function

- For each pixel, if red<128, we set red=0, otherwise we set red=255
- Similarly for green, blue
Assignment: Create a python function for each of the following:

1. Decrease red by 20%
2. Decrease green by 20%
3. Decrease blue by 20%
4. Increase red by 20%, if possible (i.e., if it does not exceed 255)
5. Similarly for increasing blue and green
6. “Posterize”
7. Think of another way to change an image and implement a function to do that