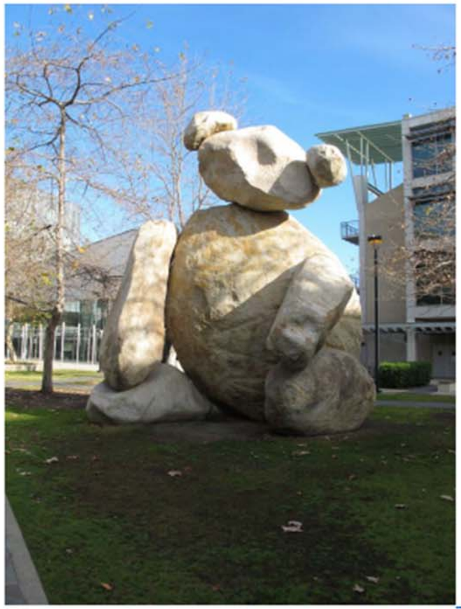
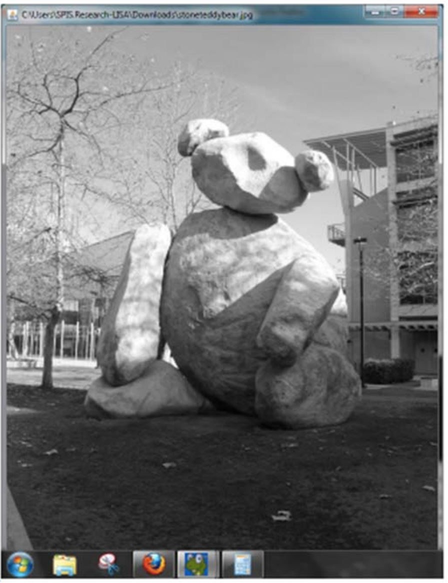
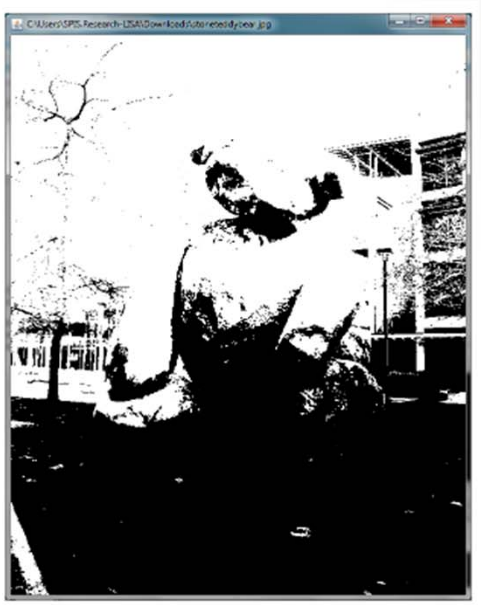
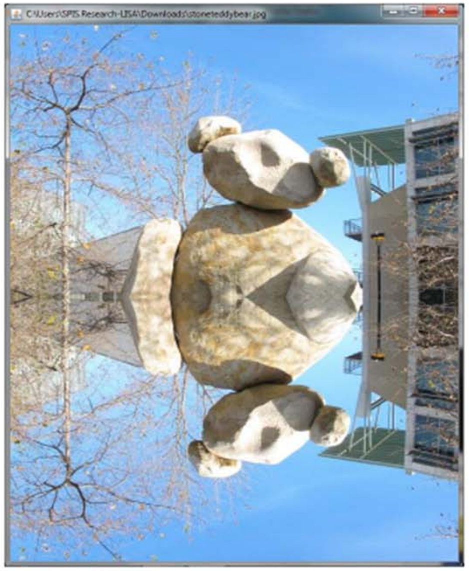
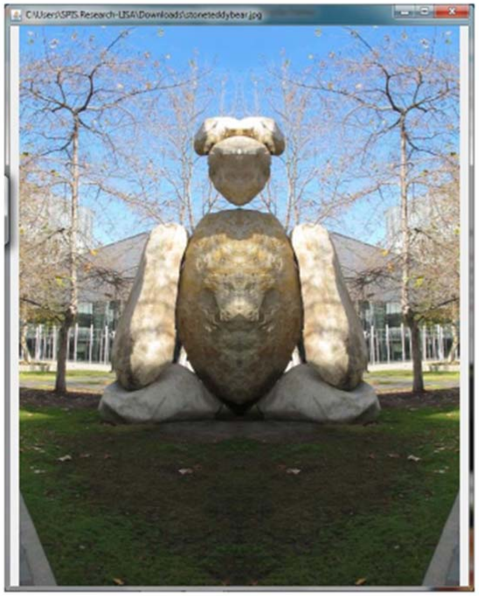
Pictures and Loops

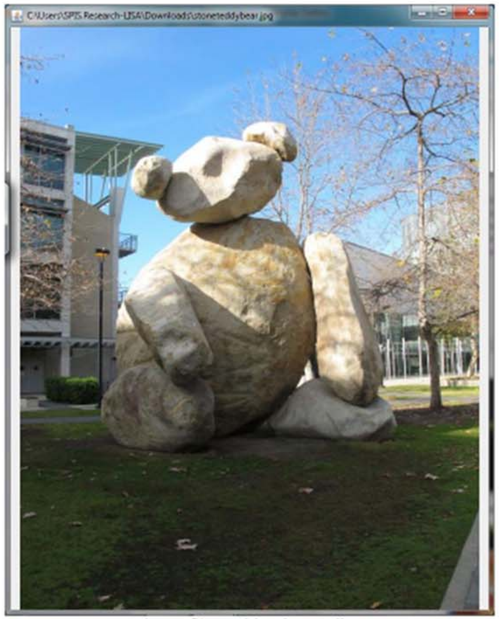






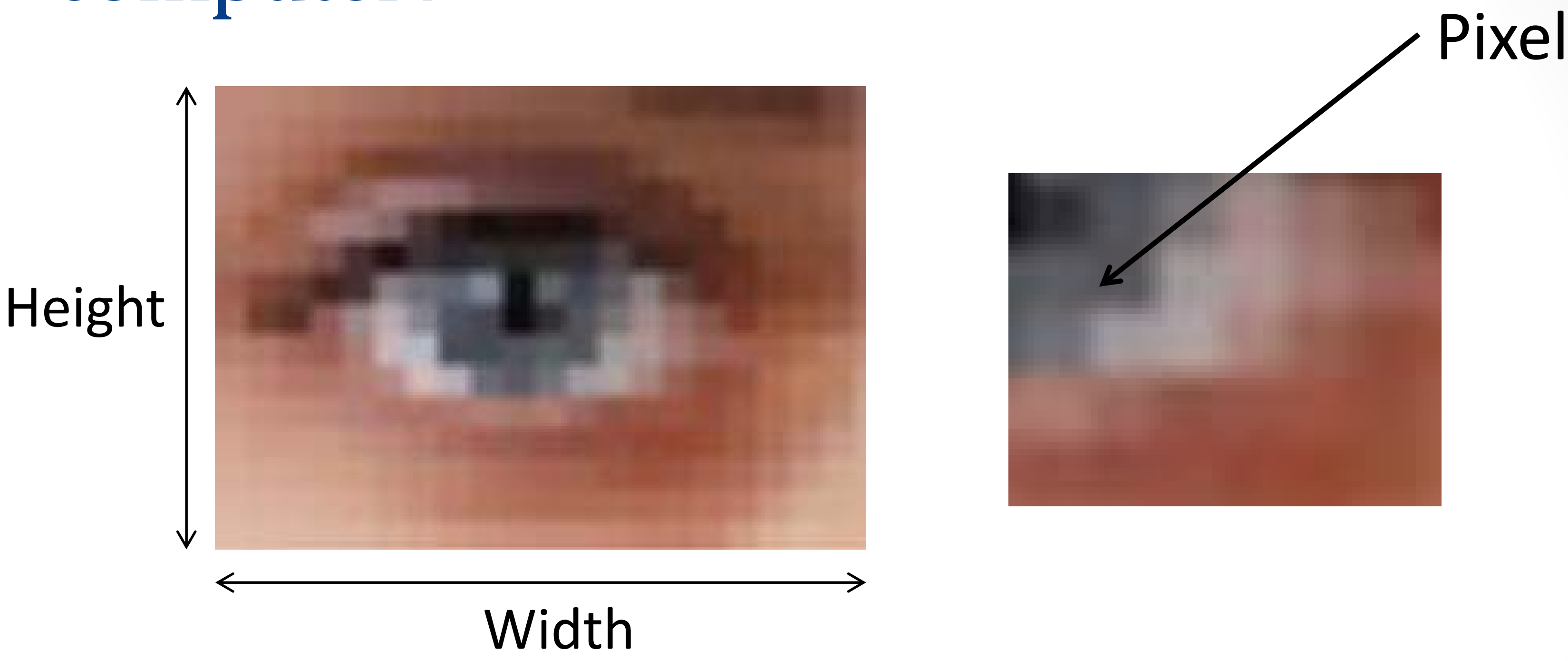




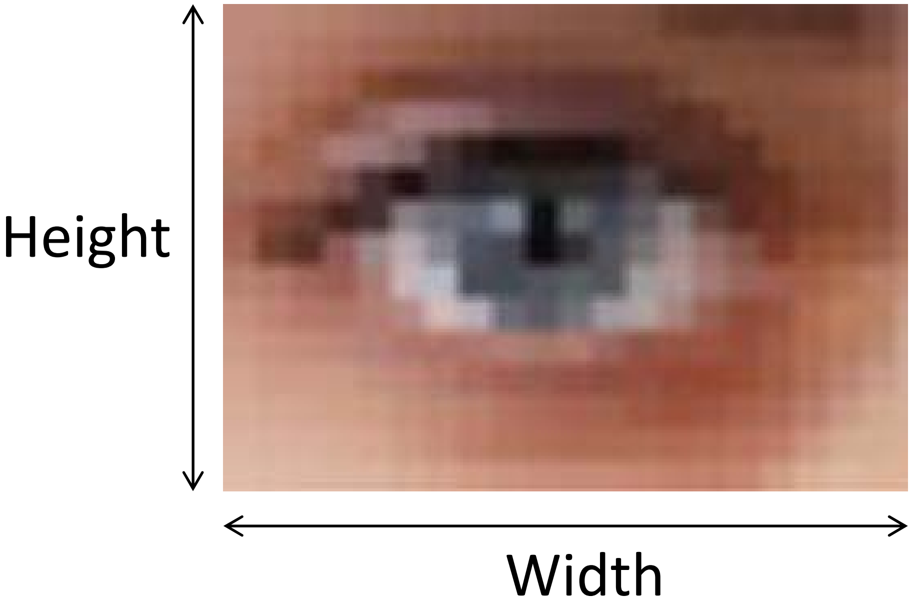


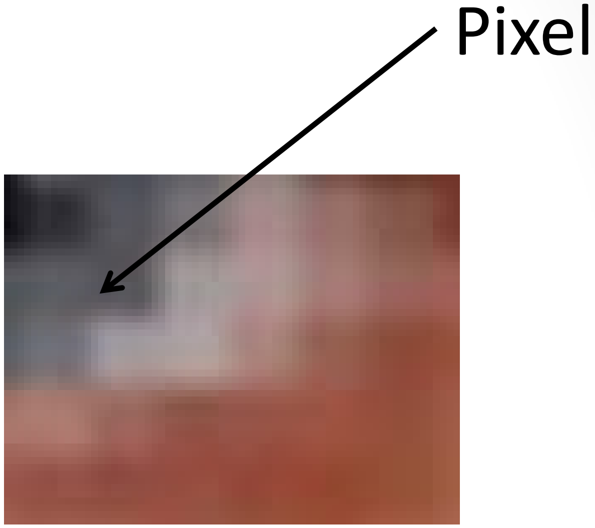
been flipped horizontally

How are pictures represented on a computer?



How are pictures represented on a computer?

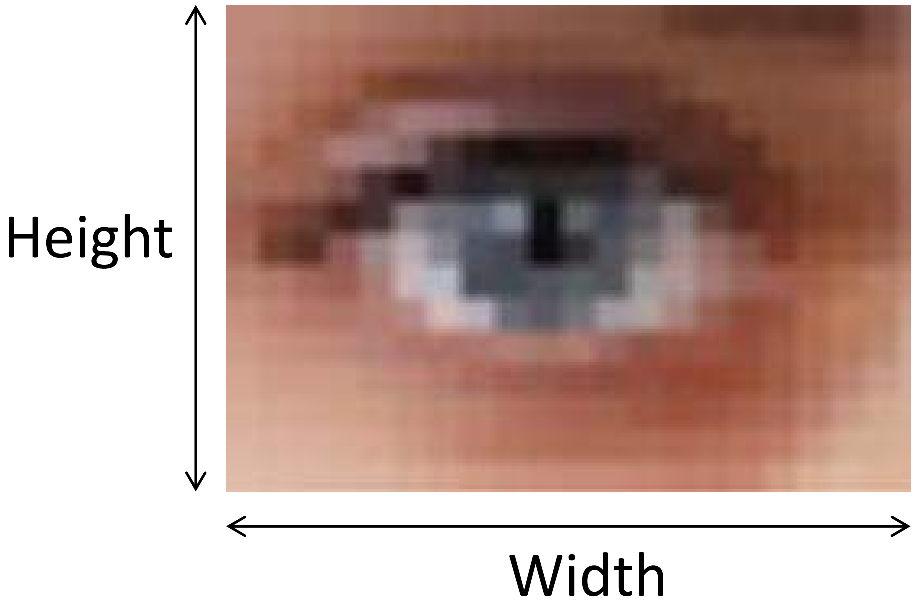


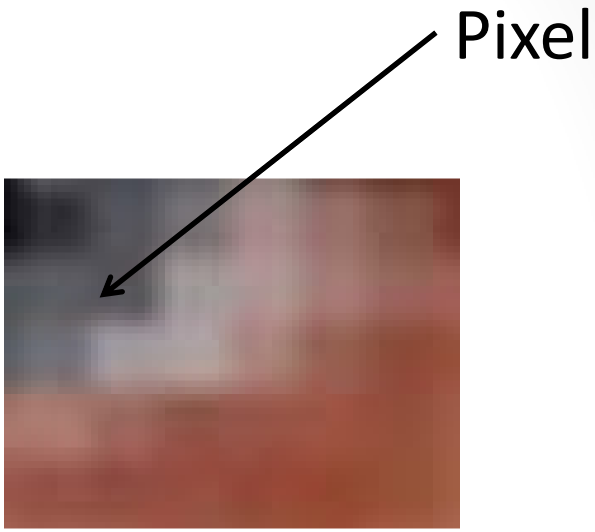


16 x 9 aspect ratio

1280 x 720	HD (720p)
1920 x 1080	Full HD (1080p)
3840 x 2160	Ultra HD, 4K

How are pictures represented on a computer?

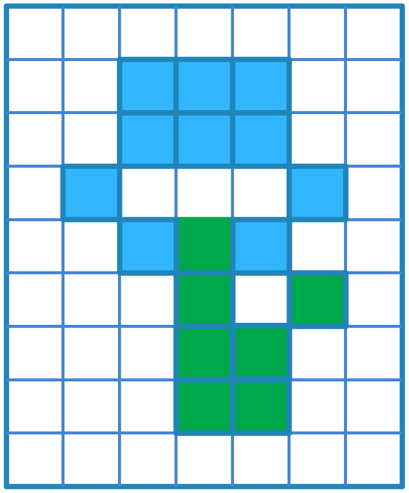




16 x 9 aspect ratio

1280 x 720	HD (720p)	1M pixels
1920 x 1080	Full HD (1080p)	2M pixels
3840 x 2160	Ultra HD, 4K	8M pixels

Each Pixel is a single Color… so how is color represented?



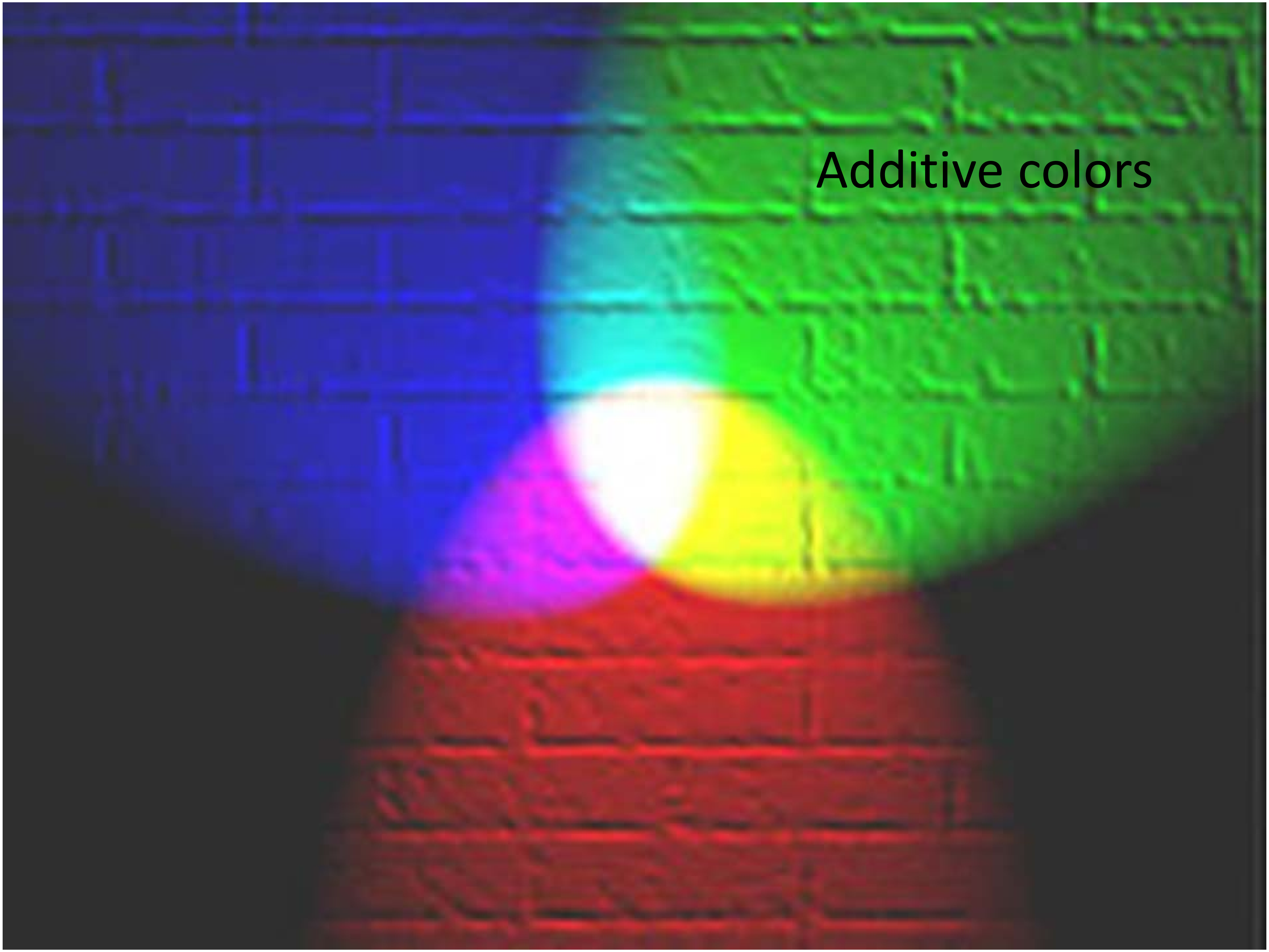
RGB Model for color representation

A color is made up of:

- Some amount of Red (0 … 255)
- Some amount of Green (0 … 255)
- Some amount of Blue (0 … 255)

Together these three channels, when combined, describe the entire range of visible colors

E.g. (R,G,B) = (102, 37, 78)



Additive colors

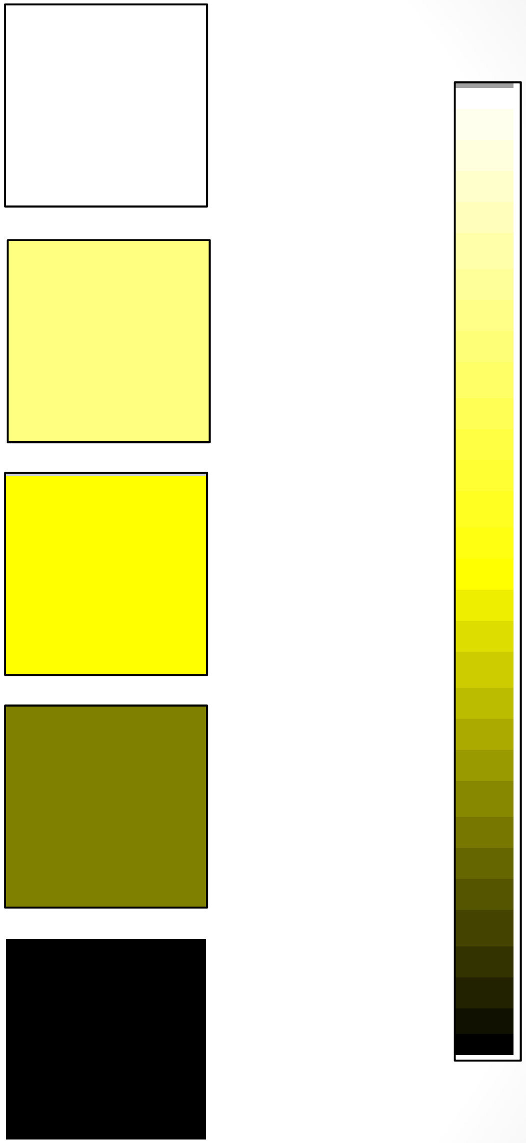
(R,G,B) = (255, 255, 255)

(R,G,B) = (255, 255, 127)

(R,G,B) = (255, 255, 0)

(R,G,B) = (127, 127, 0)

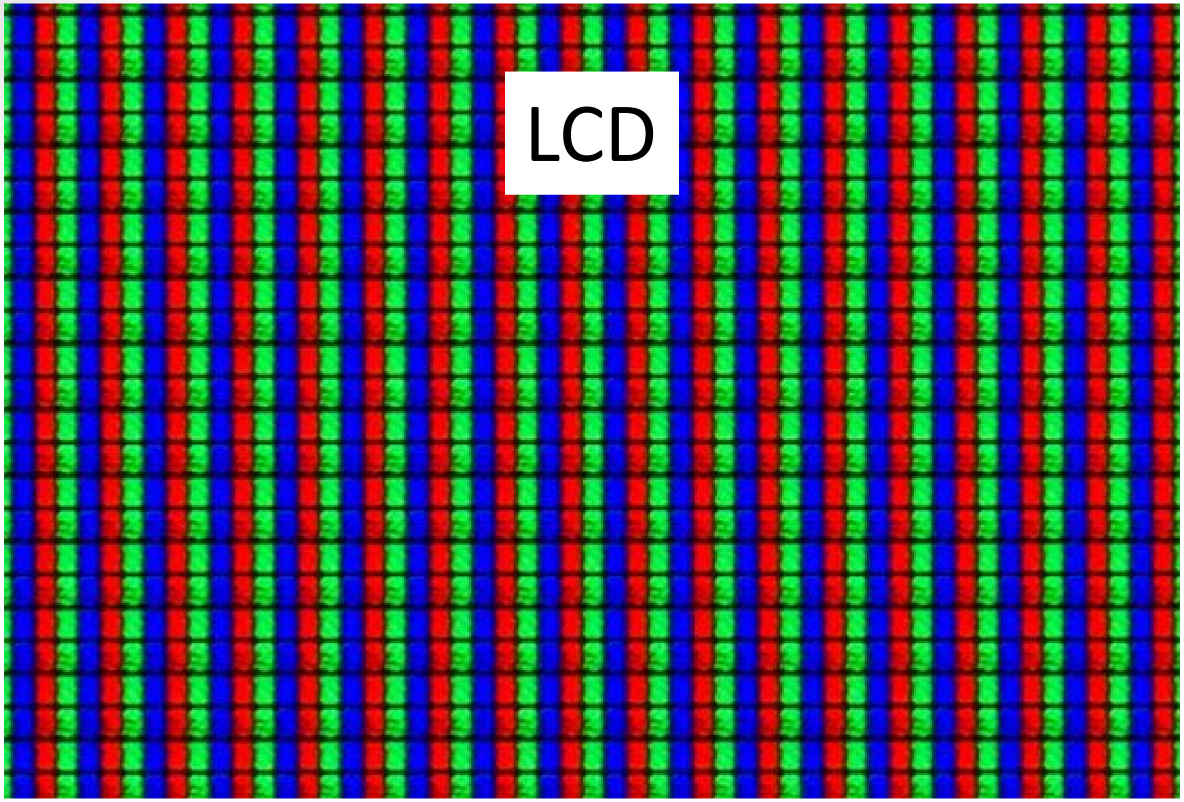
(R,G,B) = (0, 0, 0)

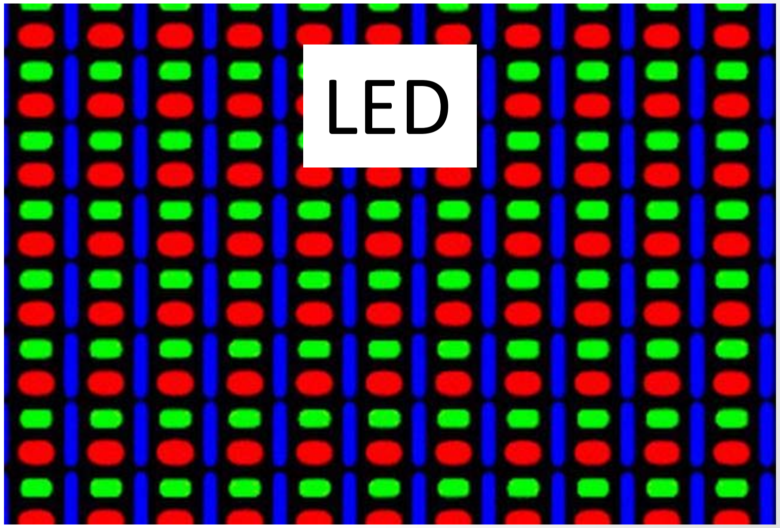


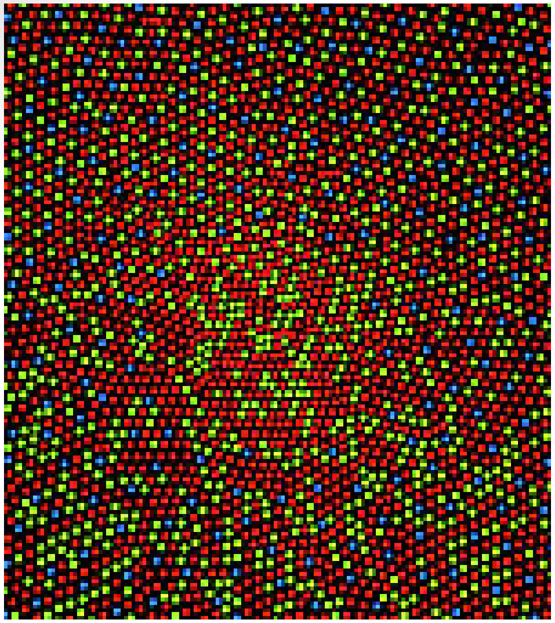
You can play with this in MS Paint, for example

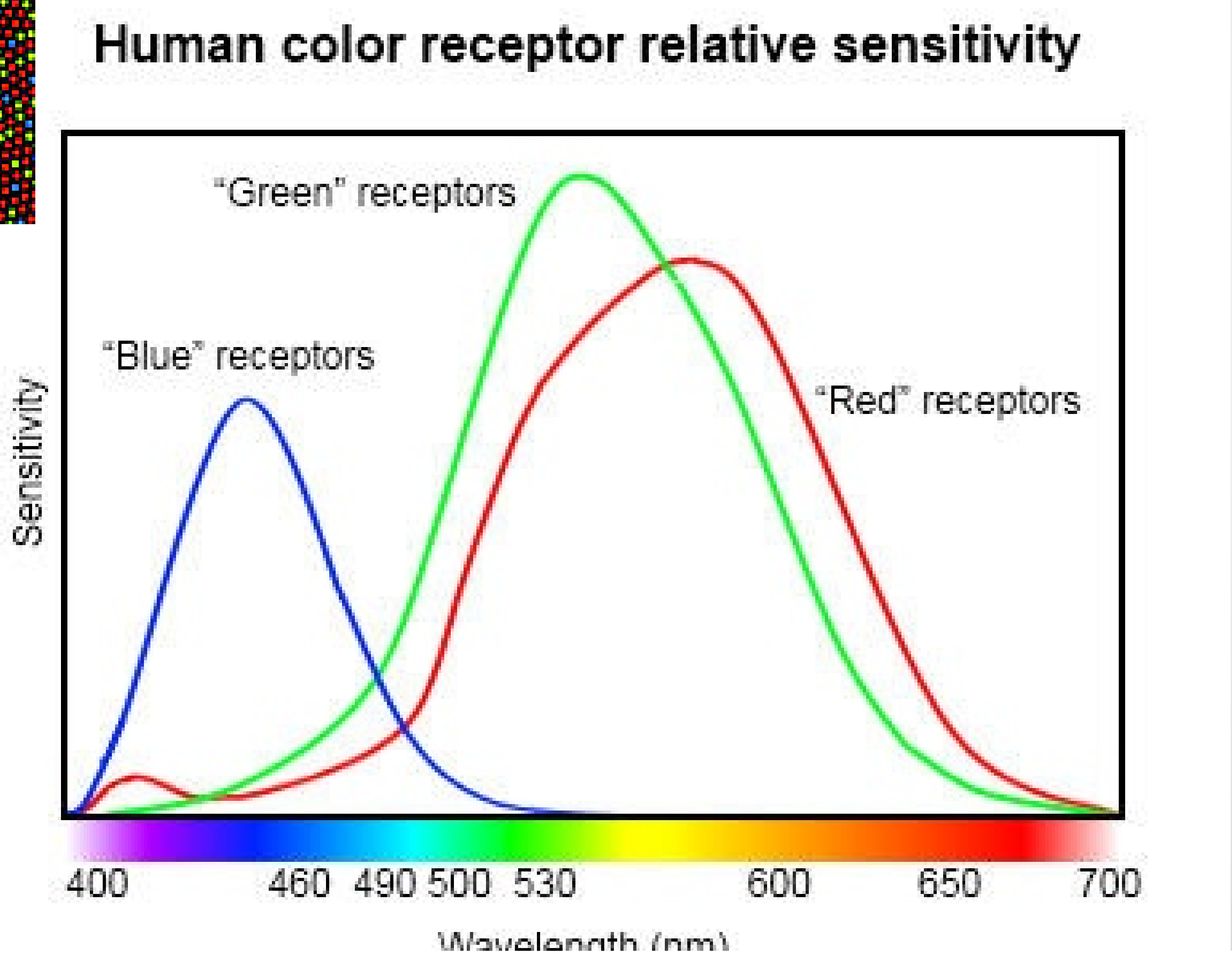
What color is represented by (100, 100,100)?

A. Black
B. White
C. Brown
D. Gray
E. Salmon


LCD


LED





Human color receptor relative sensitivity

Python Imaging Library

```
>>> from PIL import Image
```

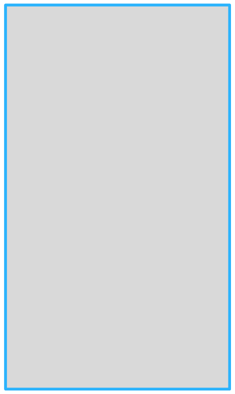
Python Imaging Library

test01

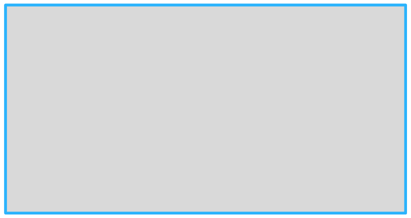
```
>>> from PIL import Image


                    mode  (width,height)    color
>>> pic = Image.new('RGB', (200,200), (0, 0, 0))


>>> pic.show()
>>> from PIL import Image
>>> pic = Image.new('RGB’, (300,600), (200, 200, 200))
>>> pic.show()
```

Which of the following is displayed?

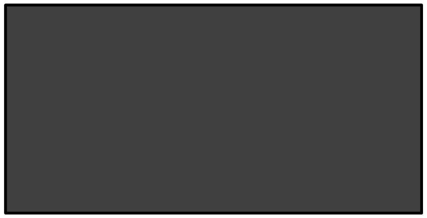
A.



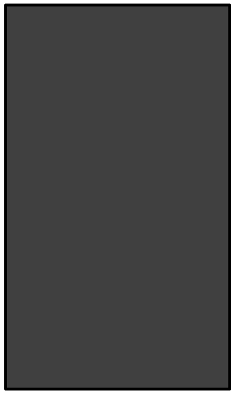
B.



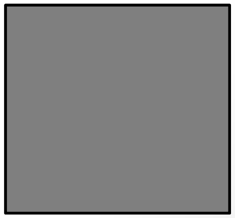
C.



D.



E.



Opening an existing picture

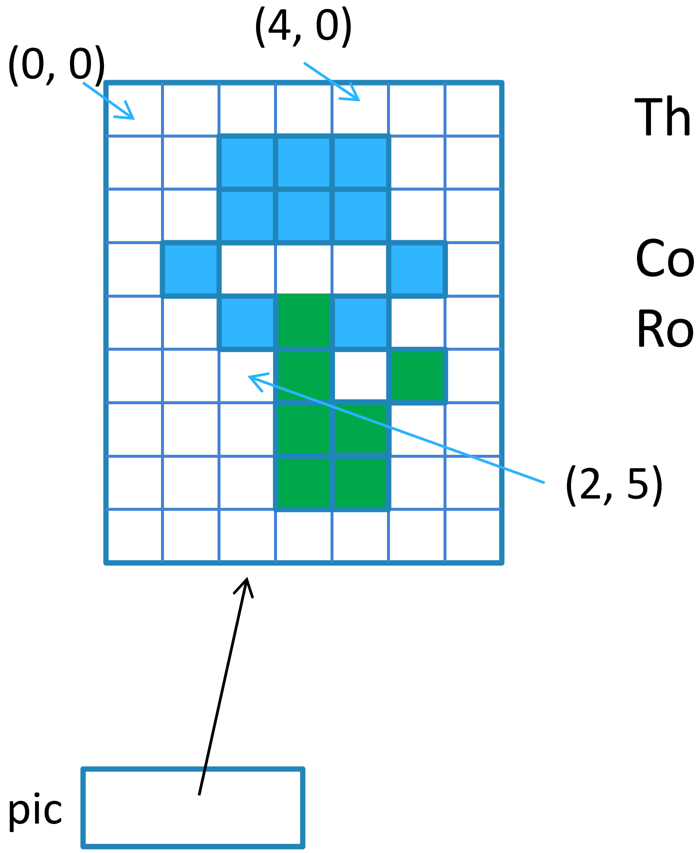
```
>>> pic = Image.open(“homerprof.jpg”)
>>> pic.show()
```

Opening an existing picture

```
>>> pic = Image.open(“homerprof.jpg”)
>>> pic.show()


>>> (w, h) = pic.size



>>> pic.save(“homerprof2.jpg”)
```

pic.size is a variable associated with the Image object.
It is a tuple with two elements: (width, height)

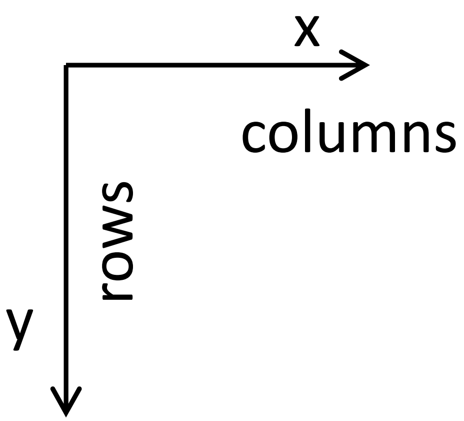
Accessing Pixels in a Picture



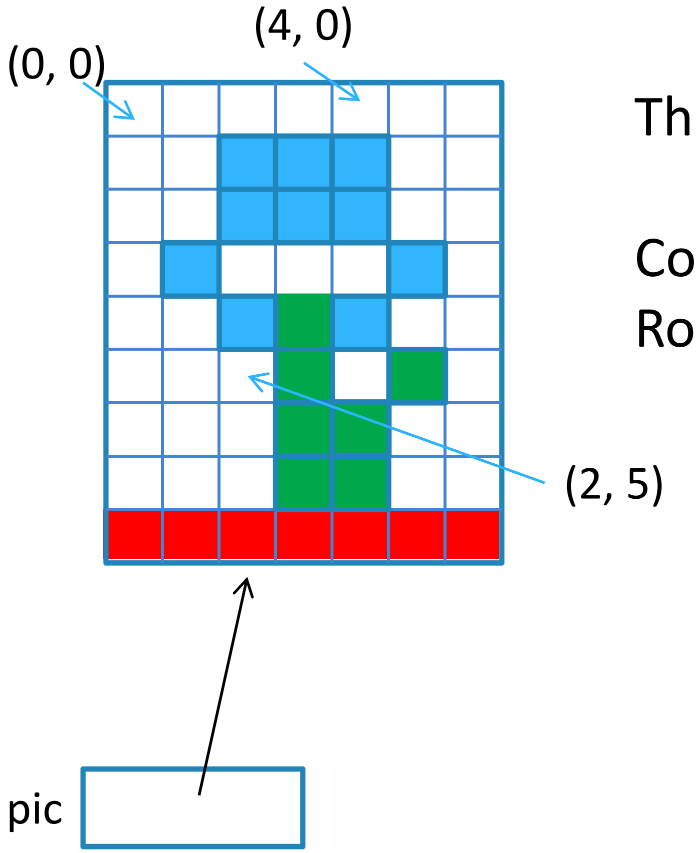
Each pixel can be accessed via its row and column

The pixel in the **upper left** is at row **0**, column **0**.

Columns increase to the right (i.e. x axis)
Rows increase ***down*** (i.e. y axis)



Accessing Pixels in a Picture



Each pixel can be accessed via its row and column

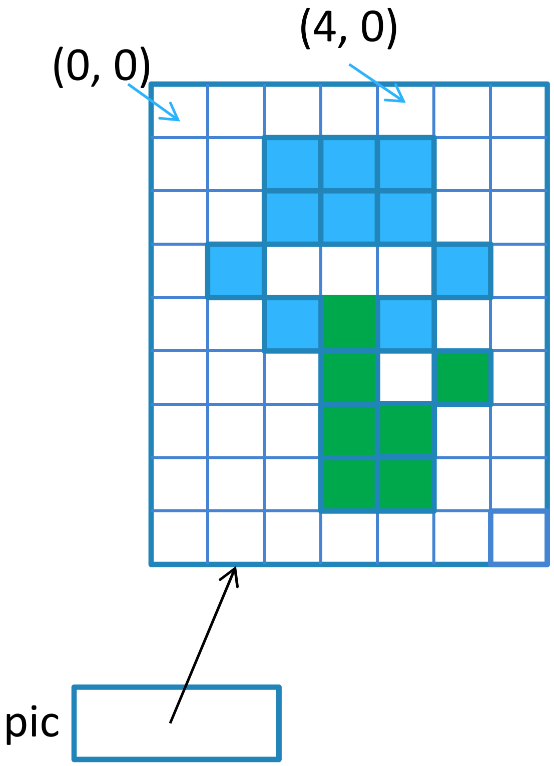
The pixel in the **upper left** is at row **0**, column **0**.

Columns increase to the right (i.e. x axis)
Rows increase ***down*** (i.e. y axis)

What value represents the last *row* of any picture, pic?

A. 0
B. pic.size[0]
C. pic.size[1]
D. pic.size[0]-1
E. pic.size[1]-1

Accessing Pixels in a Picture

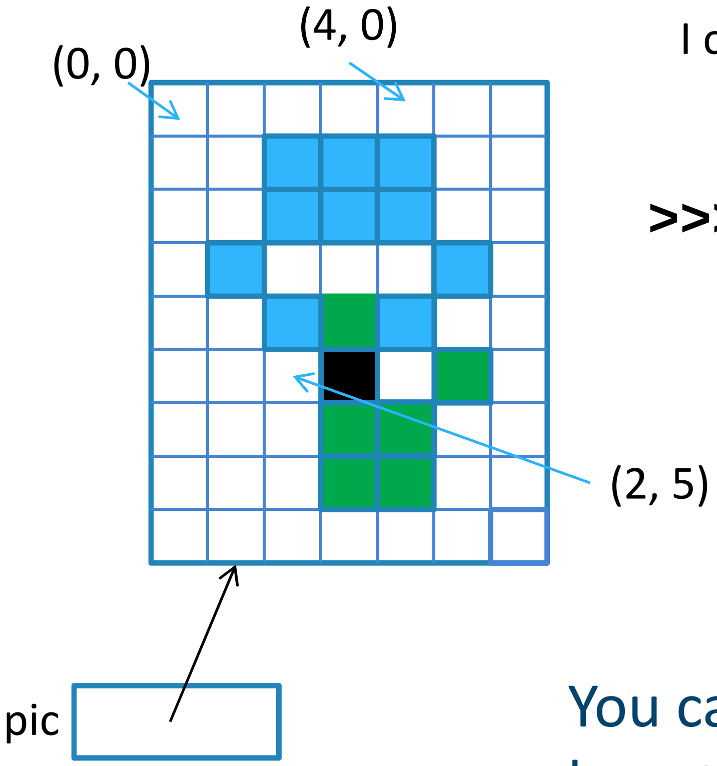


You can retrieve (the color values of) a single pixel

```
>>> pix = pic.getpixel( (3, 5) )
>>> pix
```

A. (255,255,255)
B. (255,0,0)
C. (0,255,0)
D. (0, 0, 255)
E. None of the above.

Modifying Pixels in a Picture

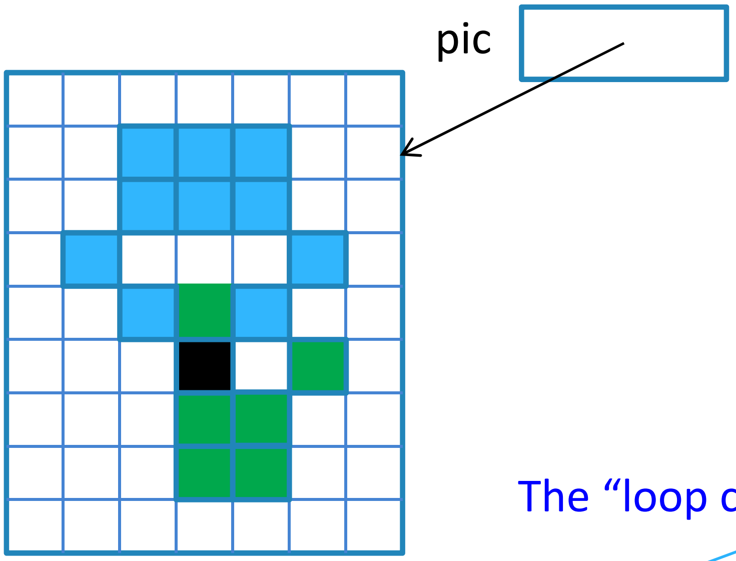


I can set the color of a pixel based on its coordinates:

```
                    x         y
>>> pic.putpixel( (3,5), (0,0,0) )
```

You can programmatically modify a picture by retrieving individual pixels and changing their color! The key is to know *which pixels* to change and *what colors* to change them to…

Loops for pixel modification



The “loop control variable”

Keyword “for”

Keyword “in”

list

```
for x in [0, 1, 2, 3, 4, 5, 6]:

        pic.putpixel( (x,0),(0,0,0))
```

Loop body

Loops

```
for x in [0, 1, 2, 3, 4, 5, 6]:
    print(x)
```

The loop body will execute one time for each element in the list. Each time through the loop, the loop control variable will take the value of the next element in the list.

Loops

```
for x in [0, 1, 2, 3, 4, 5, 6]:
    print(x)
```

```
for x in range(7):
    print(x)
```

Loops

```
for x in [0, 1, 2, 3, 4, 5, 6]:
    print(x)
```

```
for x in range(7):
    print(x)
```

```
>>> list(range(7))
[0, 1, 2, 3, 4, 5, 6]

>>> list(range(1, 7))
[1, 2, 3, 4, 5, 6]

>>> list(range(0, 7, 2))
[0, 2, 4, 6]
```

Draw a line

```
for x in [0, 1, 2, 3, 4, 5, 6]:
    pic.putpixel( (x,0),(100,100,100))
```

```
for x in range(7):
    pic.putpixel( (x,0),(0,0,0))
```

```
pic.putpixel( (0,0),(0,0,0))
pic.putpixel( (1,0),(0,0,0))
pic.putpixel( (2,0),(0,0,0))
pic.putpixel( (3,0),(0,0,0))
pic.putpixel( (4,0),(0,0,0))
pic.putpixel( (5,0),(0,0,0))
pic.putpixel( (6,0),(0,0,0))
```