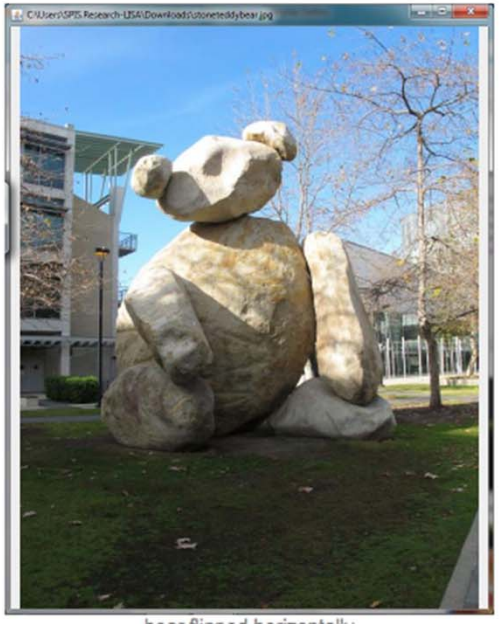
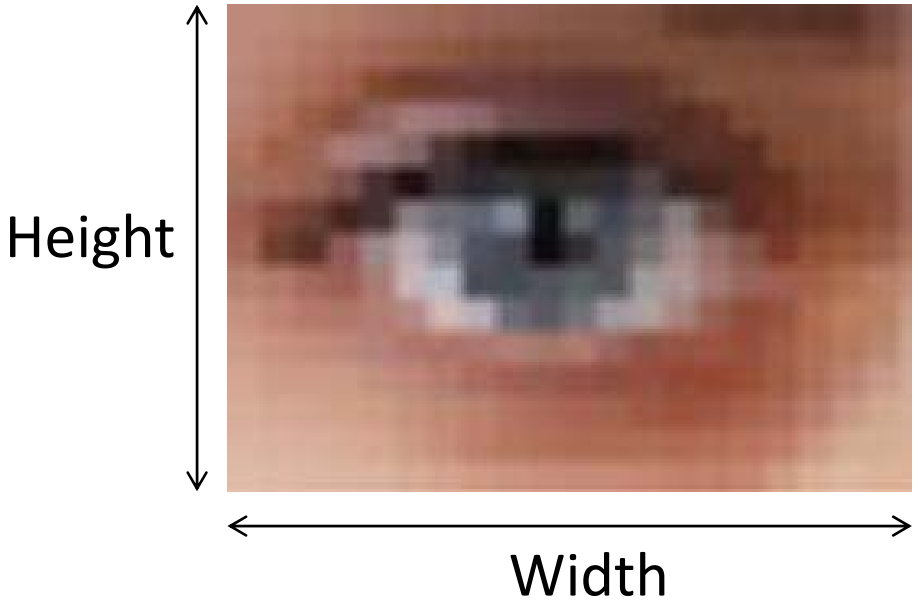
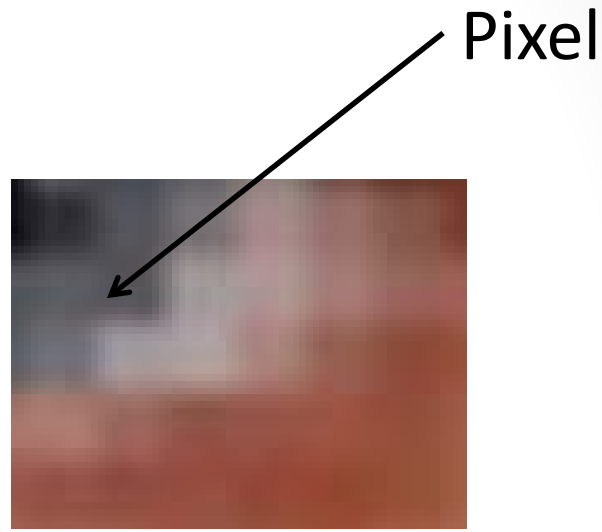
Images



How are images represented on a computer?

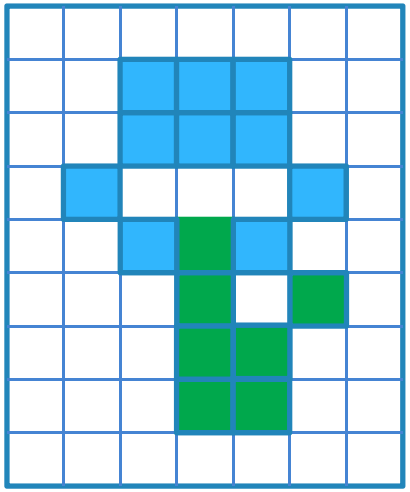




16 x 9 aspect ratio

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?



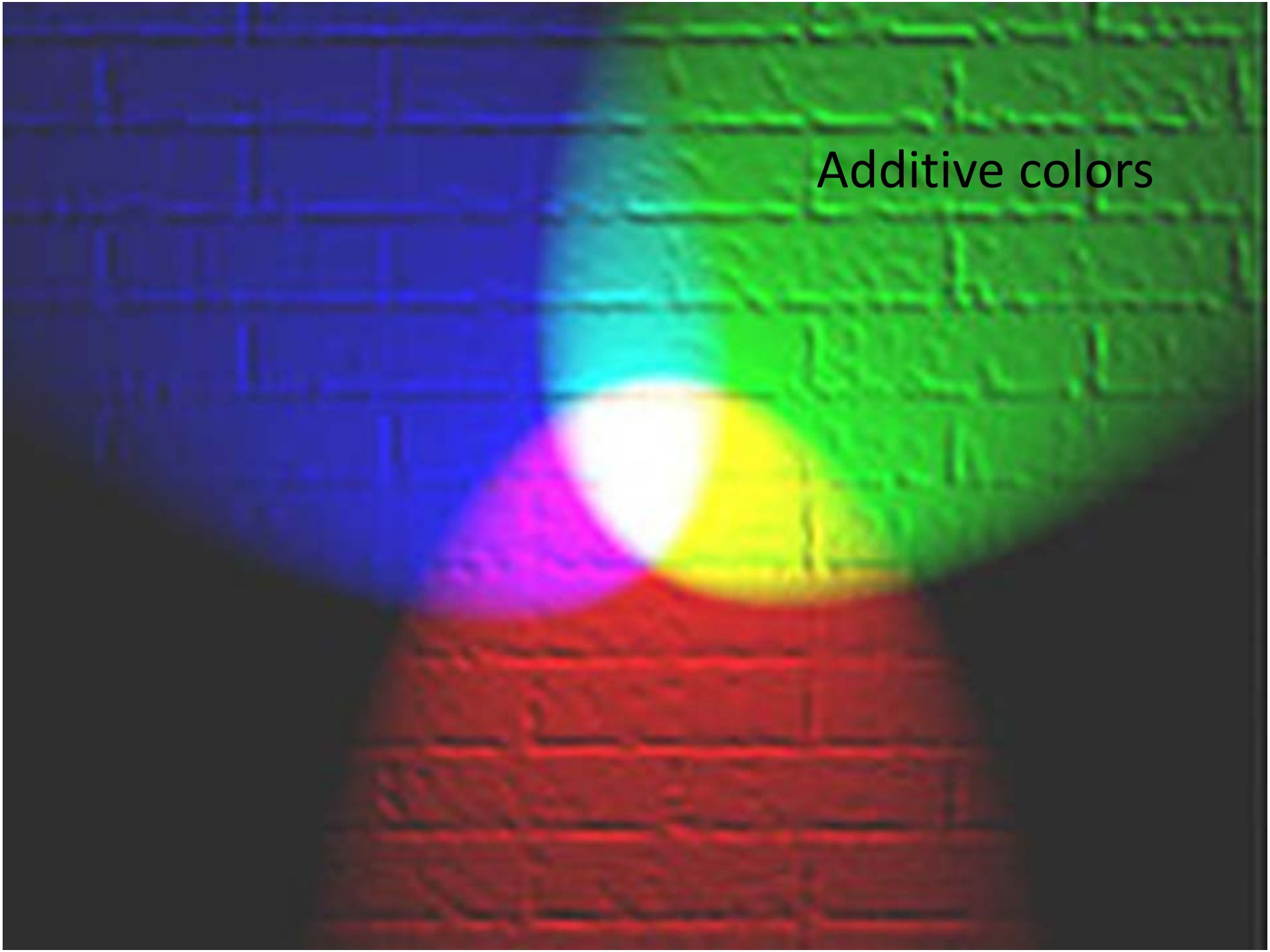
RGB Model for color representation

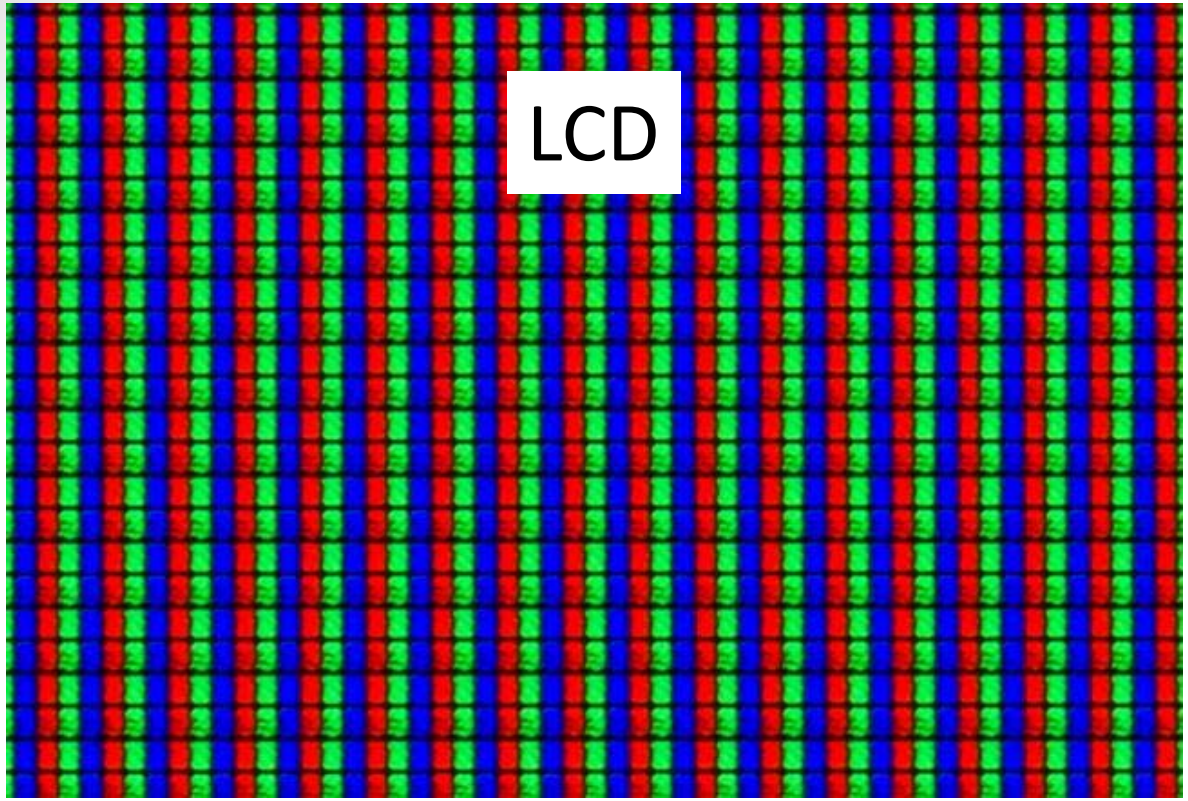
A color is made up of:

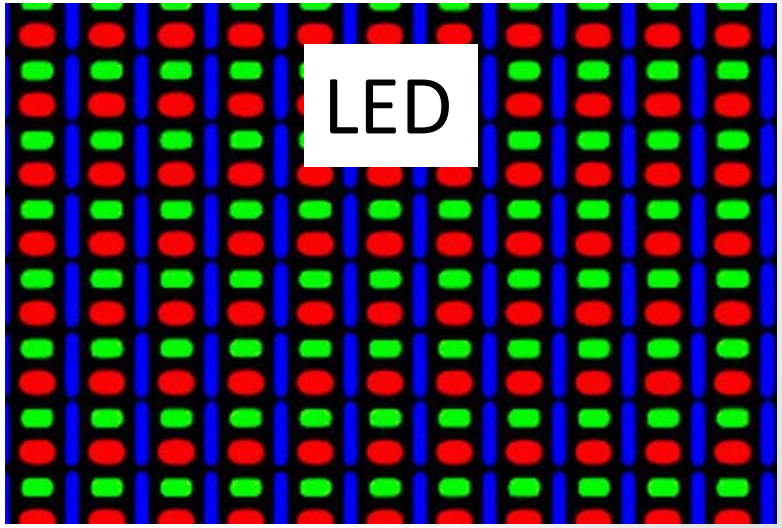
- Some amount of Red (0 … 255)
- Some amount of Green (0 … 255)
- Some amount of Blue (0 … 255)

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

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


Additive colors


LCD


LED

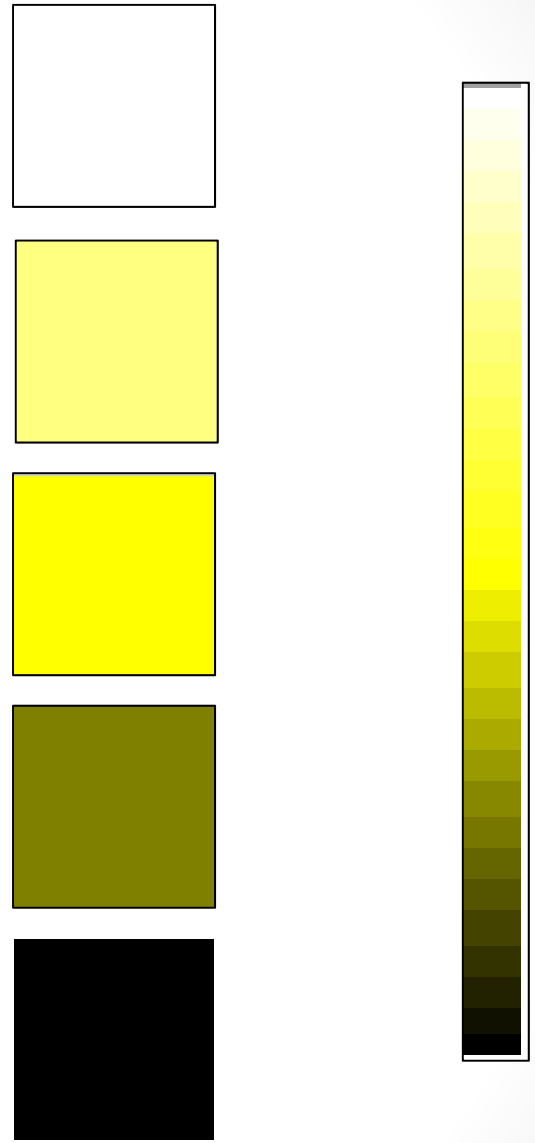
(R,G,B) = (255, 255, 255)

(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

Python Imaging Library

```
from PIL import Image
```

Python Imaging Library

test01

```
from PIL import Image
```

mode (width,height) color

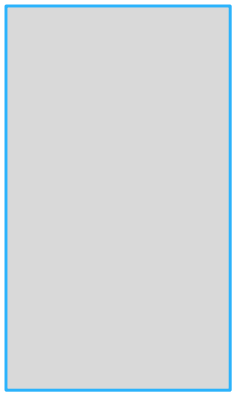
```
pic = Image.new('RGB', (200,200), (0, 0, 0))
```

```
pic.show()
```

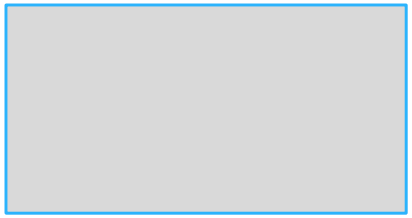
```
from PIL import Image
pic = Image.new('RGB', (300,600), (200, 200, 200))
pic.show()
```

Which of the following is displayed?

A.



B.



C.

D.

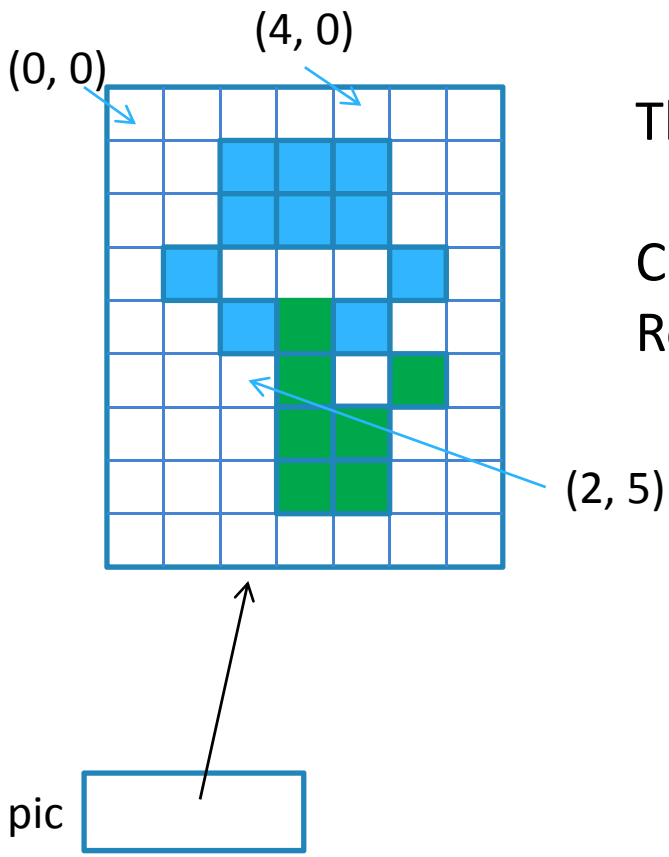
E.

Opening an existing picture

```
pic = Image.open("homerprof.jpg")
pic.show()

(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)

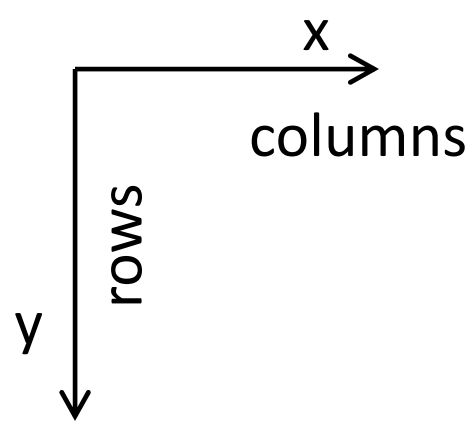
Accessing Pixels in a Picture



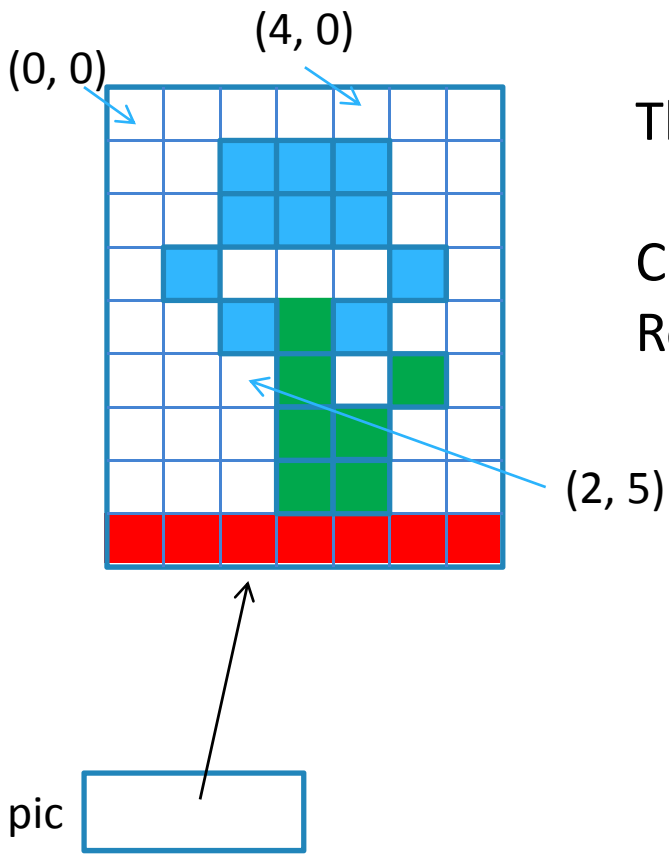
Each pixel can be accessed via its row and column

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

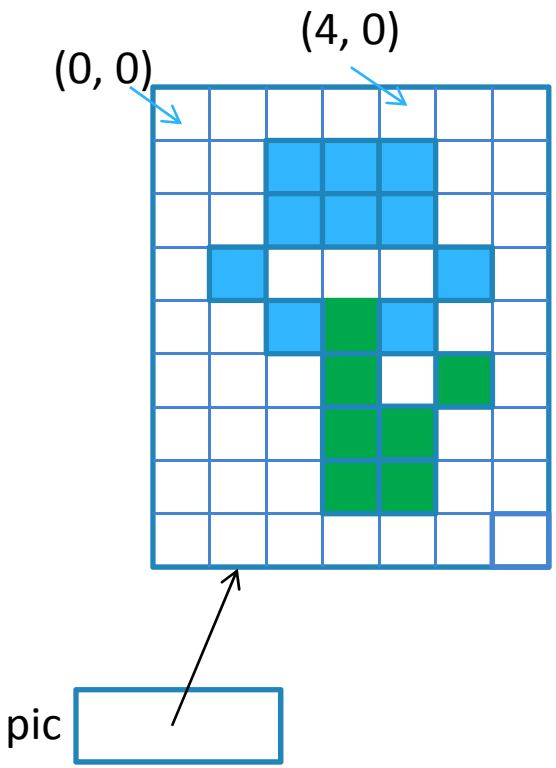
The pixel in the **upper left** is at row **0**, column **0**.

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



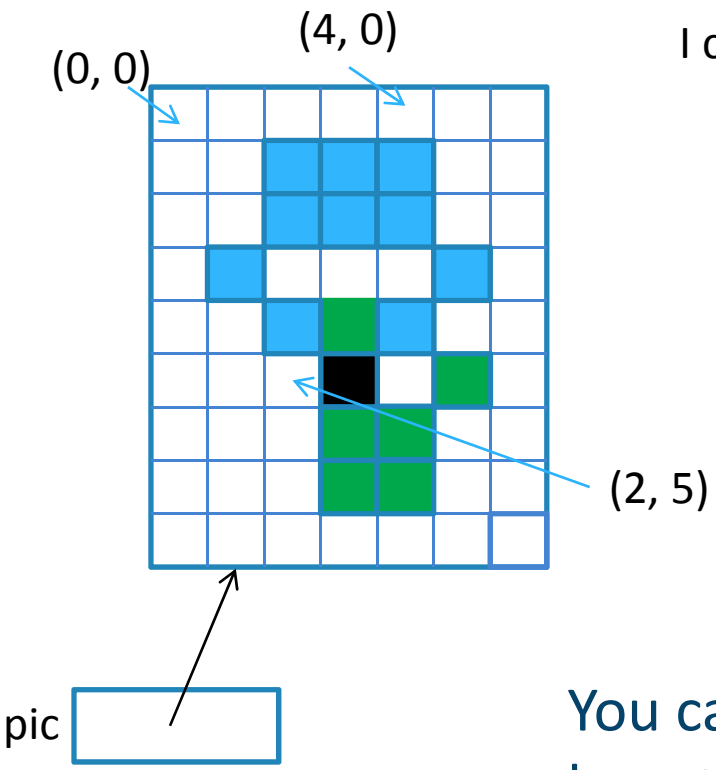
You can retrieve (the color values of) a single pixel

```
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

test02



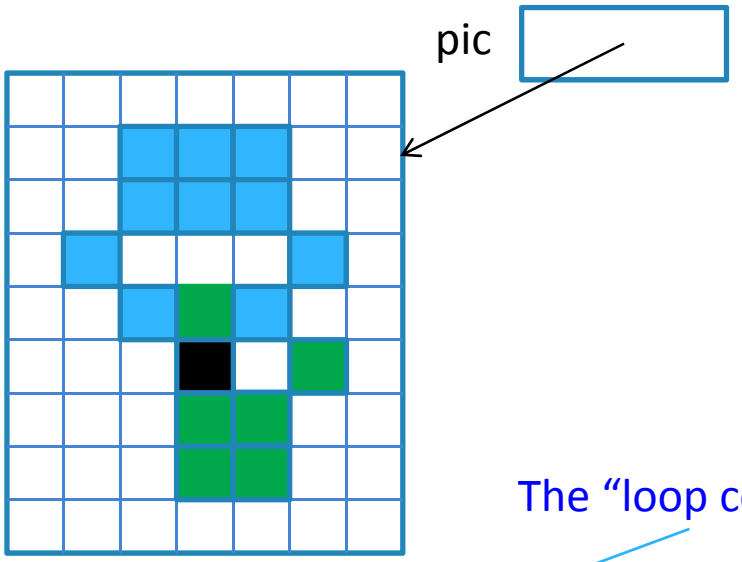
I can set the color of a pixel based on its coordinates:

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

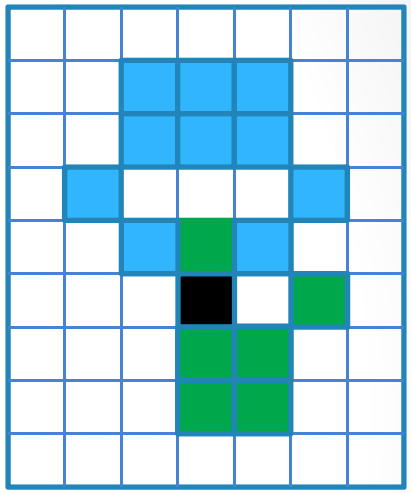
```
for x in [0, 1, 2, 3, 4, 5, 6]:

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

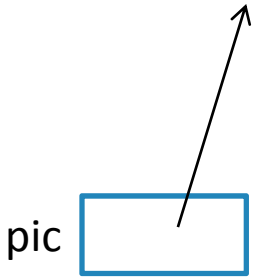
Loop body

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))
```

Draw a line

How can we draw a complete horizontal line across any image?

A.
```
for x in range(pic.size[0]):
    pic.putpixel( (x,2),(255,0,0))
```

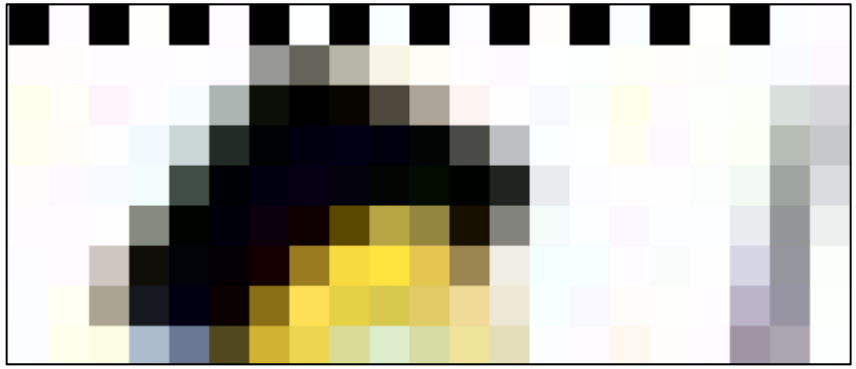
B.
```
for x in range(pic.size[0]-1):
    pic.putpixel( (x,2),(255,0,0))
```

```
pic.size == (21,30)
```

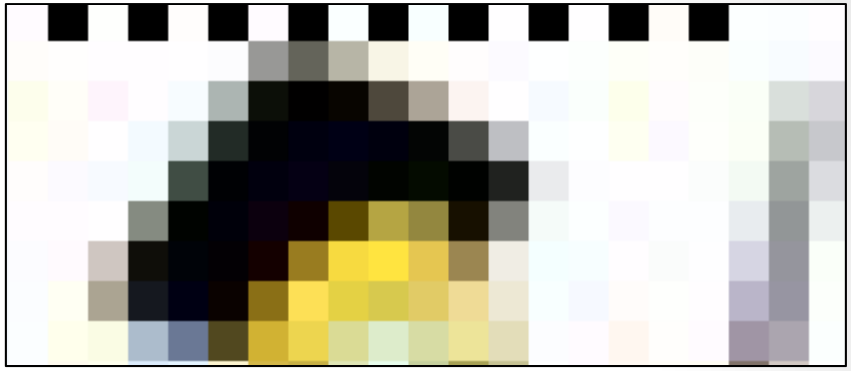
test03

```
for x in range(1,19,2):
    pic.putpixel((x,0),(0,0,0))
```

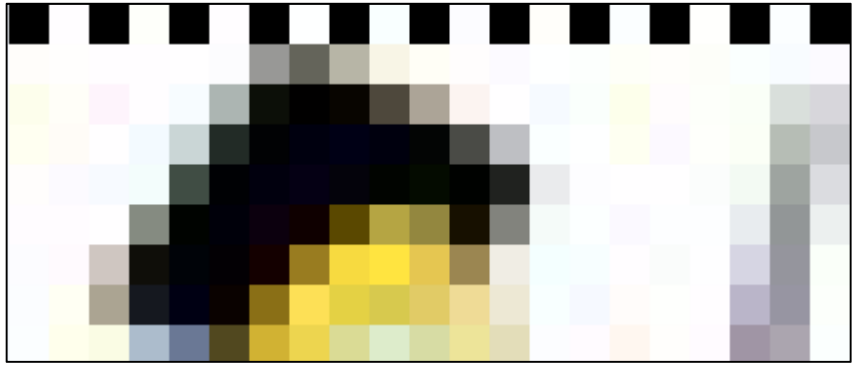
A.



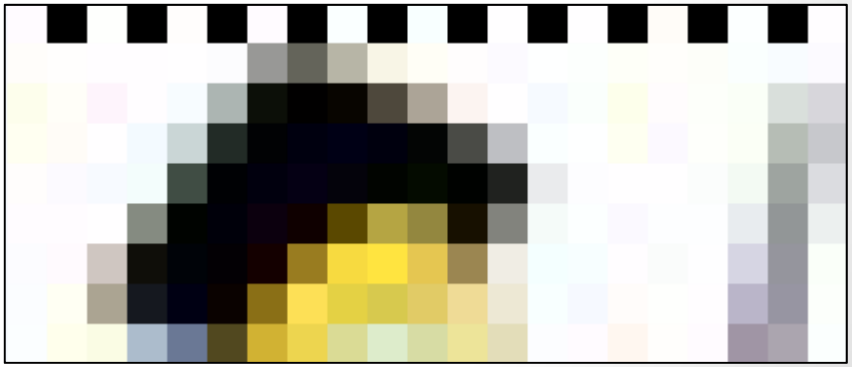
B.



C.



D.



E. Something else

Nested loops

```
for x in range(2):
    for y in range(2):
        …
```

Nested loops

```
for x in [1, 2]:
    for y in [3,4]:
        print(y)
```

What will the output look like?

A.	B.	C.	D.	E.
1 2 3 4	3 4	3 3 4 4	3 4 3 4	3 4 4

Nested loops

```
for y in [1, 3]:
    for x in [2,4]:
        print(x,y)
```

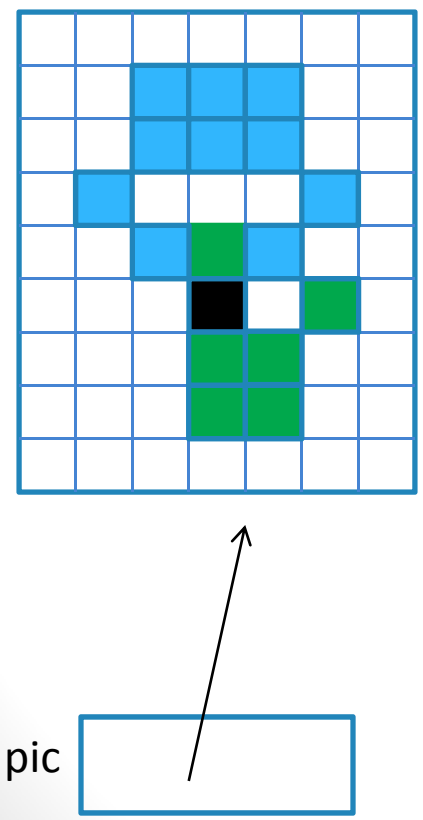
What will the output look like?

A.	B.	C.	D.	E.
2 1	2 1	1 2	1 2	2 1
2 3	4 1	1 4	3 2	4 3
4 1	2 3	3 2	1 4	2 3
4 3	4 3	3 4	3 4	4 1

Nested loops for modifying the whole image

test04

```
for x in range(pic.size[1]//2):
    for y in range(pic.size[0]):
      pic.putpixel( (x,y), (100,100,100))
```



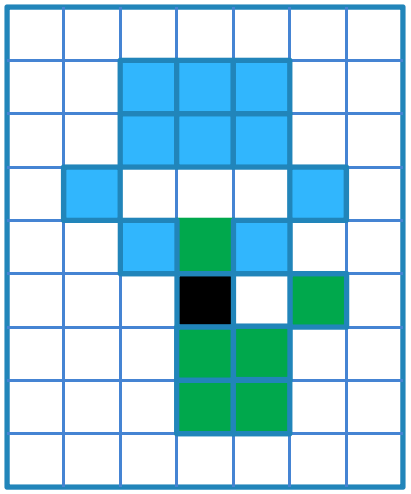
What does the code above do?

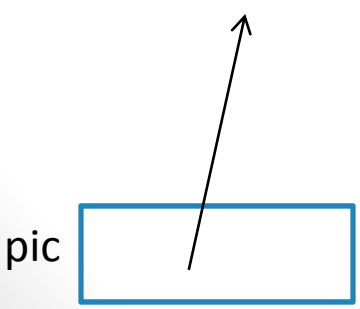
A. Turns the top half of the picture gray
B. Turns the bottom half of the picture gray
C. Turns the right half of the picture gray
D. Turns the left half of the picture gray
E. Something else

Nested loops for modifying the whole image

test04

```
for x in range(pic.size[0]//2):
    for y in range(pic.size[1]):
      pic.putpixel( (x,y), (100,100,100))
```





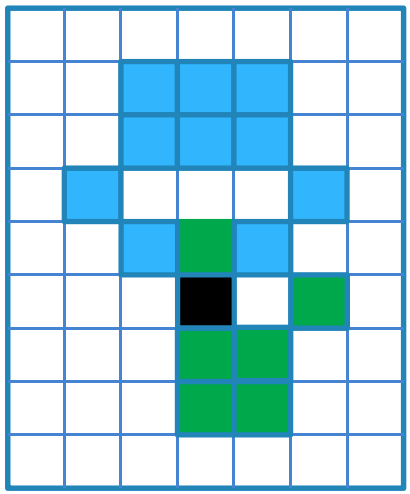
What does the code above do?

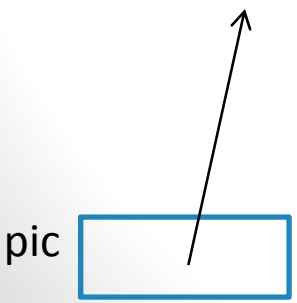
A. Turns the top half of the picture gray
B. Turns the bottom half of the picture gray
C. Turns the right half of the picture gray
D. Turns the left half of the picture gray
E. Something else

If statements in loops too

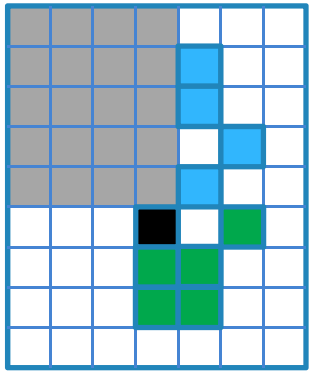
test05

```
for x in range(pic.size[0]):
    for y in range(pic.size[1]):
        if y < pic.size[1]//2 and x < pic.size[0]//2:
            pic.putpixel( (x,y), (100, 100, 100))
```

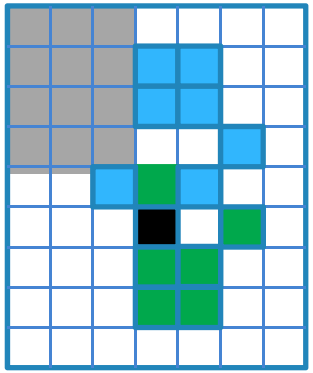




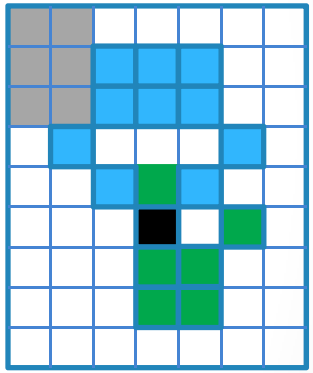
A



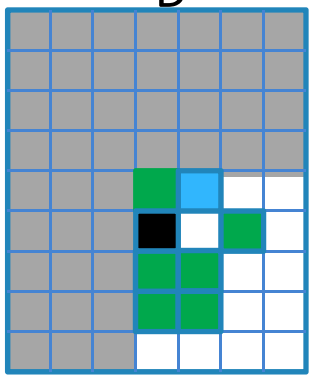
B



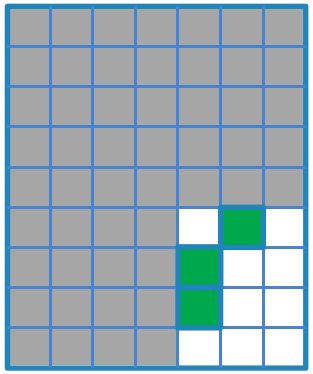
C



D



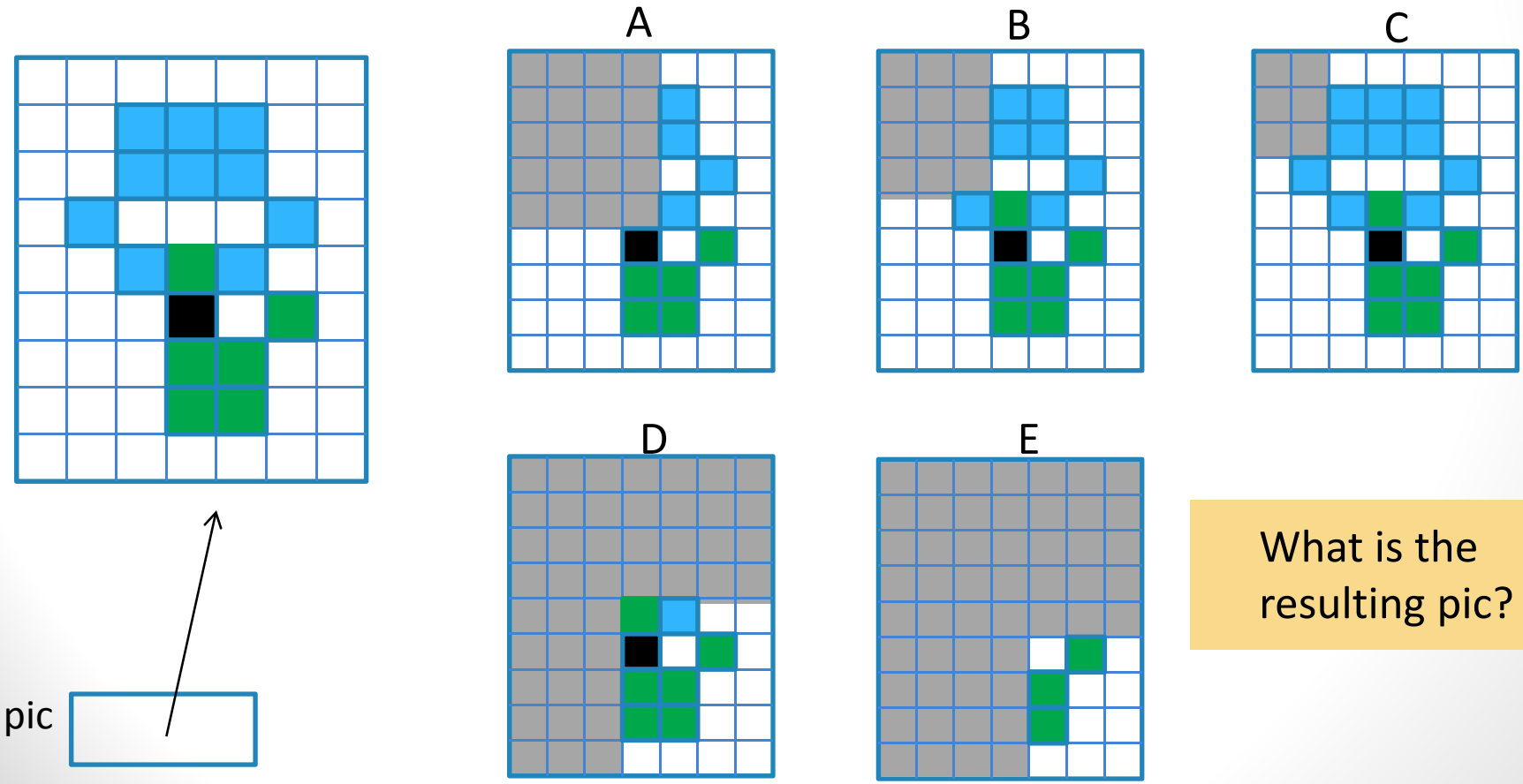
E



What is the resulting pic?

If statements work in loops too!

```
for x in range(pic.size[0]):
    for y in range(pic.size[1]):
        if y < pic.size[1]//2 or x < pic.size[0]//2:
            pic.putpixel( (x,y), (100, 100, 100))
```

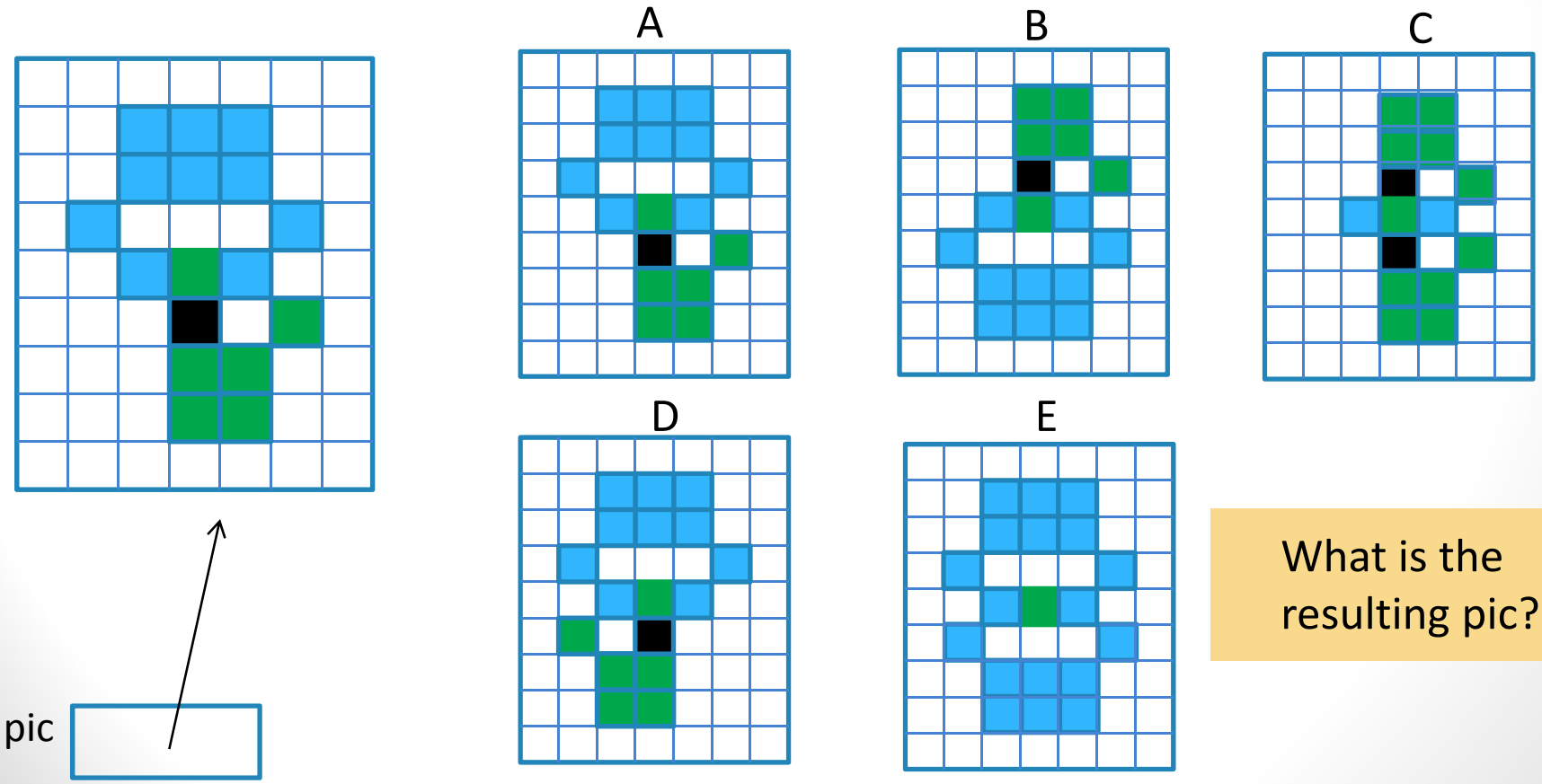


What is the resulting pic?

Flipping the image upside down

test06

```
for x in range(pic.size[0]):
   for y in range(pic.size[1]):
      (r,g,b) = pic.getpixel( (x,y) )
      pic.putpixel( (x,pic.size[1]-y-1), (r,g,b) )
```



What is the resulting pic?