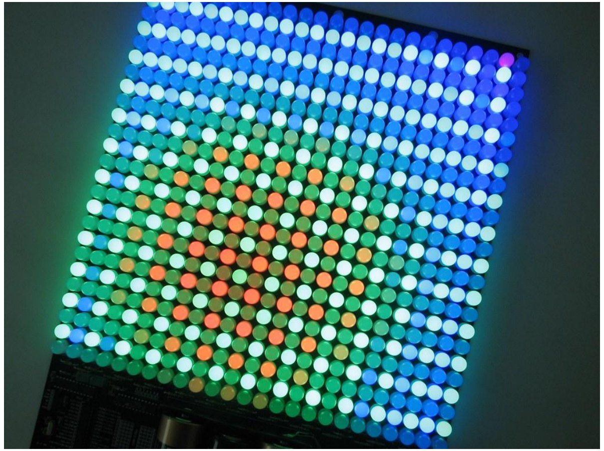
CodeX and RGB

Setting pixels to any color using RGB tuples

FIRIA LABS

What is RGB?



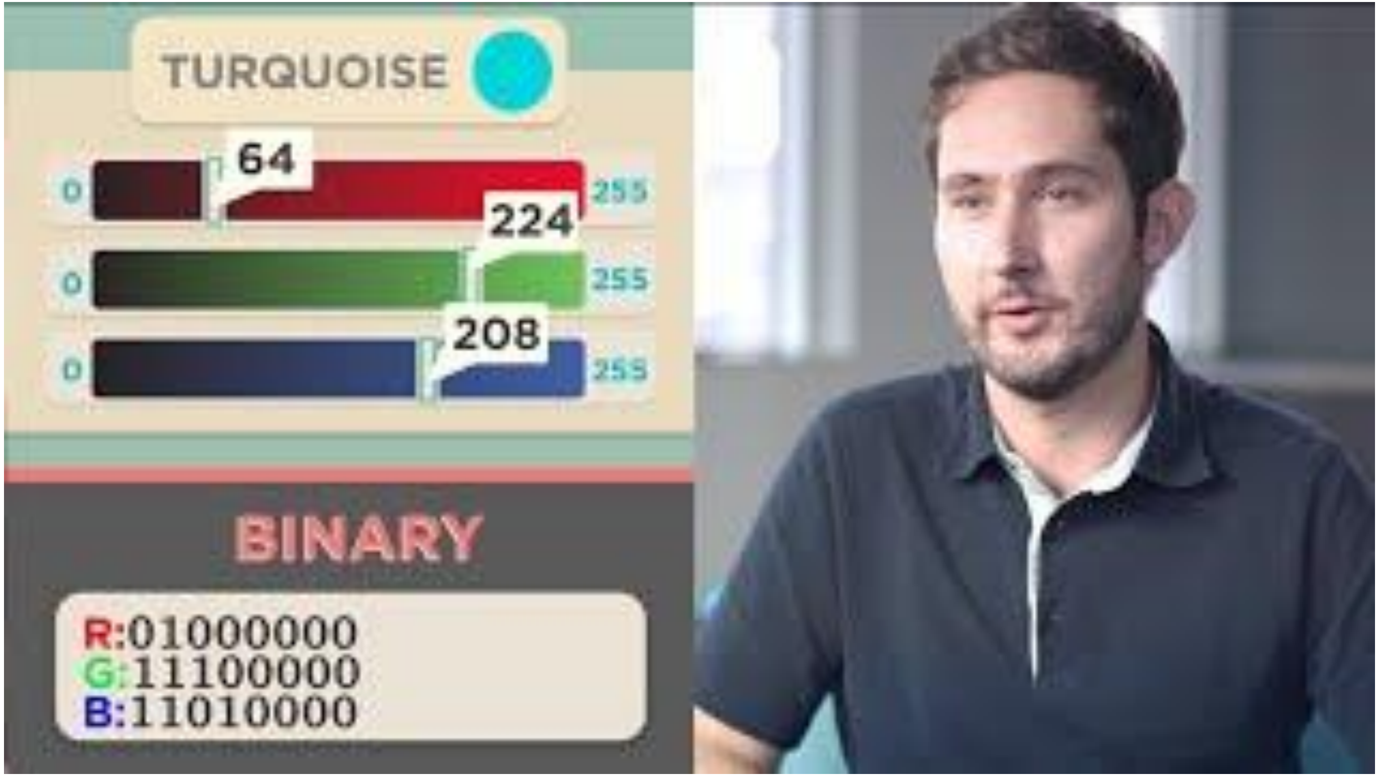
RGB stands for Red, Green, Blue.

Find out more by watching the short video on the next slide.

Code.org video on pixels

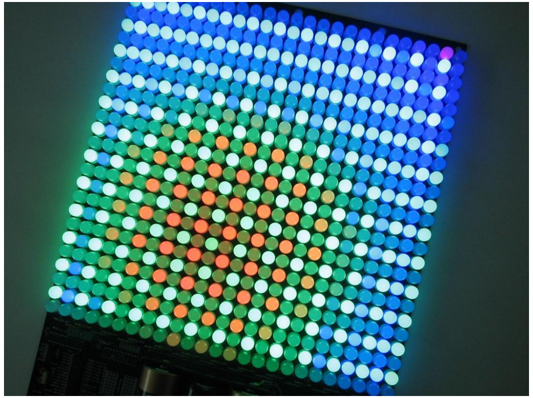
Watch from

0:45 to 2:35



FIRIA LABS

Getting RGB values

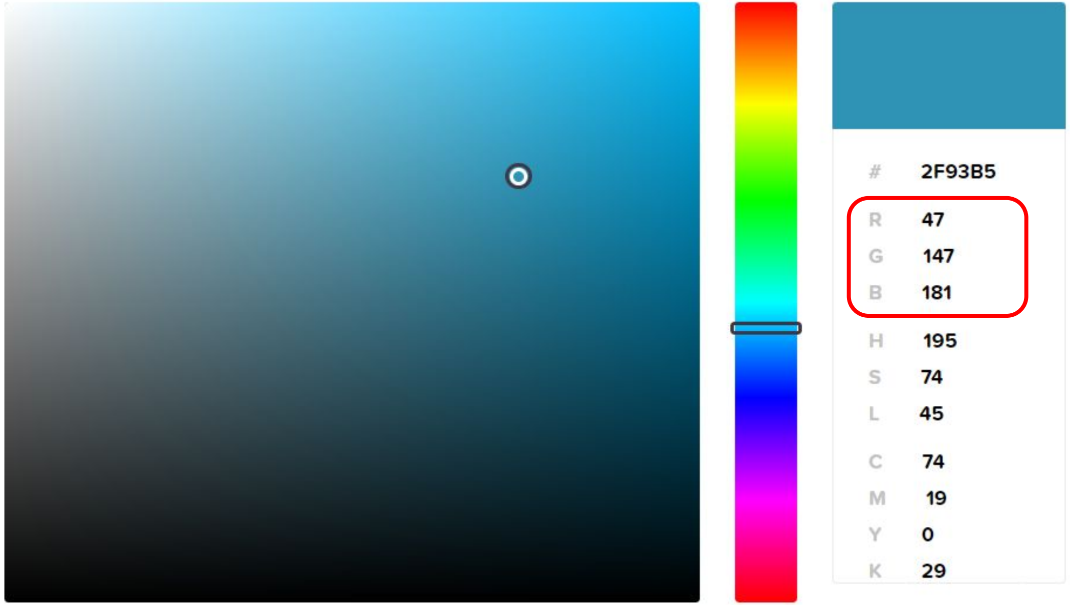


The video mentions “triplets” of numbers. Each number in the triplet represents a value in RGB.

🟦 = (**47**, **147**, **181**)

- The first number is the amount of red
- The second number is the amount of green
- The third number is the amount of blue

Getting RGB values



# 2F93B5

R 47
G 147
B 181

H 195
S 74
L 45

C 74
M 19
Y 0
K 29

Use online software to select a color and find the RGB colors.

(47, 147, 181)

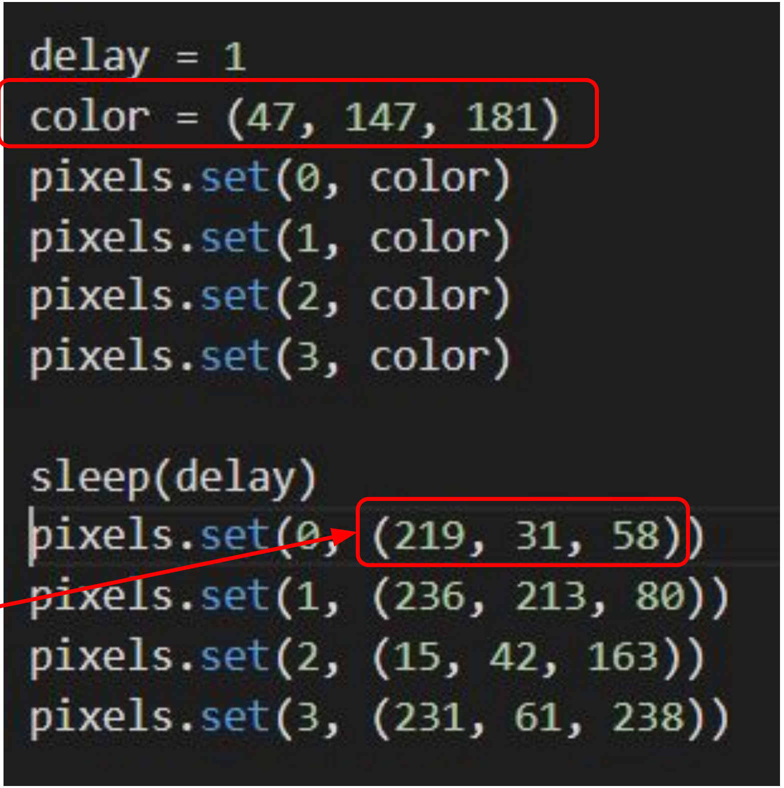
- In Python, the triplet is called a “tuple”

Using RGB values

Set your own colors by changing the color value to a tuple instead of a built-in color:

color = (47, 147, 181)

You can do this throughout the program, as often as you want. Or set the value in the pixels.set command.



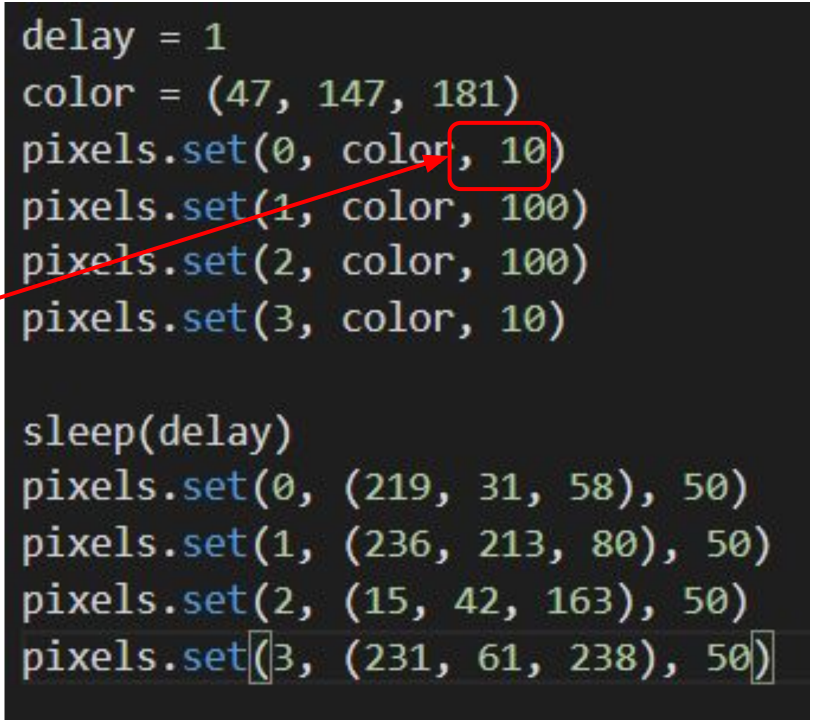
```
delay = 1
color = (47, 147, 181)
pixels.set(0, color)
pixels.set(1, color)
pixels.set(2, color)
pixels.set(3, color)

sleep(delay)
pixels.set(0, (219, 31, 58))
pixels.set(1, (236, 213, 80))
pixels.set(2, (15, 42, 163))
pixels.set(3, (231, 61, 238))
```

Using RGB values

You can add a third value in the pixels.set() command, which determines the brightness.

Brightness values are from 0 to 100



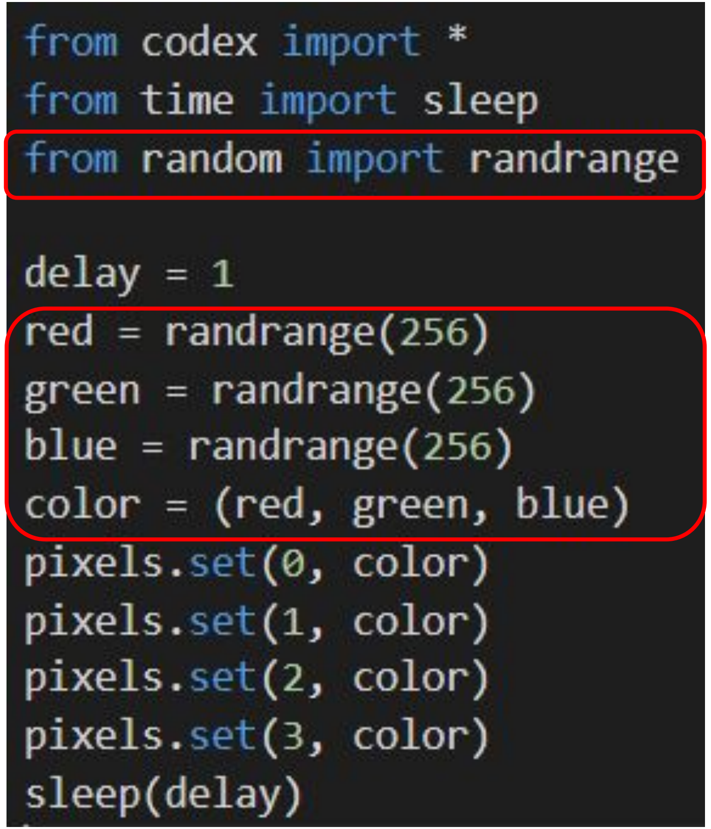
```
delay = 1
color = (47, 147, 181)
pixels.set(0, color, 10)
pixels.set(1, color, 100)
pixels.set(2, color, 100)
pixels.set(3, color, 10)

sleep(delay)
pixels.set(0, (219, 31, 58), 50)
pixels.set(1, (236, 213, 80), 50)
pixels.set(2, (15, 42, 163), 50)
pixels.set(3, (231, 61, 238), 50)
```

Random RGB values

If you want to try something new, generate random numbers for R, G and B and see what color happens!

Everytime you run the code, or add the code multiple times, you should get a different color.



```
from codex import *
from time import sleep
from random import randrange

delay = 1
red = randrange(256)
green = randrange(256)
blue = randrange(256)
color = (red, green, blue)
pixels.set(0, color)
pixels.set(1, color)
pixels.set(2, color)
pixels.set(3, color)
sleep(delay)
```

Clearing your CodeX

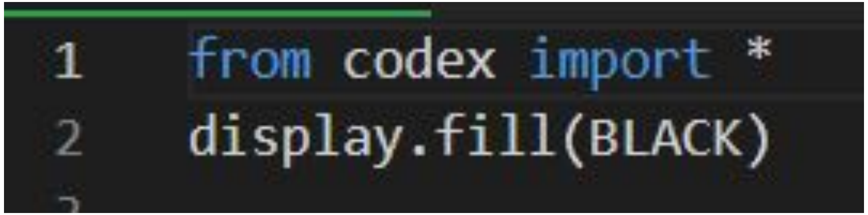
The need for clearing code

You will share the CodeX with other students.

- Every time you run your program, it is loaded onto the CodeX
- The last program executed stays on the CodeX
- Therefore, you want the last program to be something that clears the CodeX and isn’t an assignment

Create a file “Clear”

- Create a new file in CodeSpace
  - Name the file “Clear”
- Type this code:



```
from codex import *
display.fill(BLACK)
```

- Run the code
  - The CodeX should be blank, with no pictures or lights on
- Run this code at the end of every class period