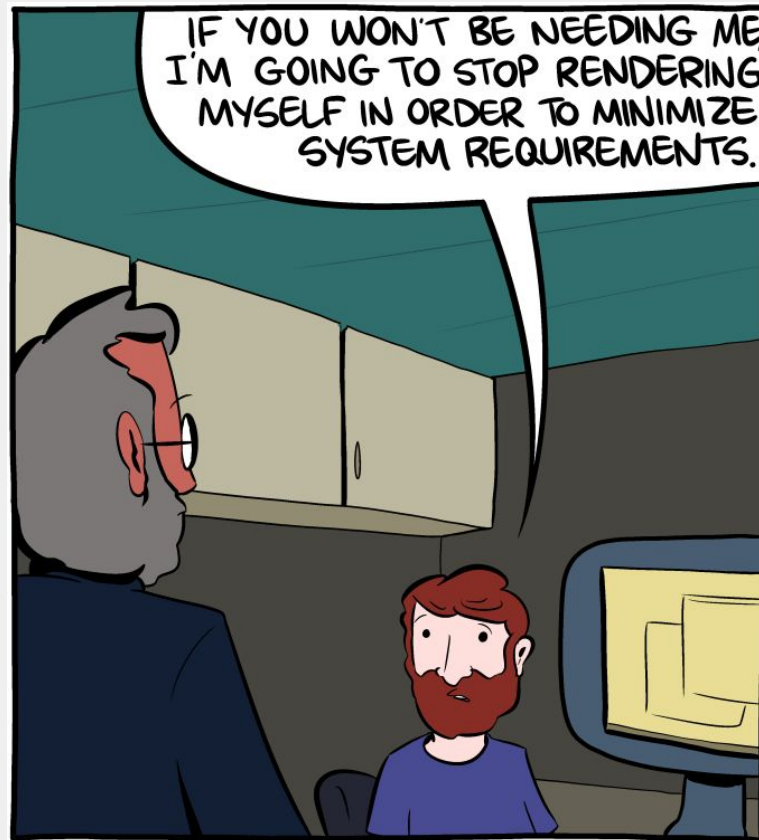


CS 110

Graphics and Motion

Adriana Picoral (she/her/hers)



Fun Fact:
Computer programmers cease to exist
when you're not looking.

Announcements

- Motion Parallax assignment due date pushed to Thursday (Oct 13)

Motion

- When you watch a movie or play a video game and see motion, this is typically a bunch of images, displayed in quick succession
- This gives the perception of motion
- When writing a graphical program that has animation, we can do so with a **draw loop**

Motion



The general outline



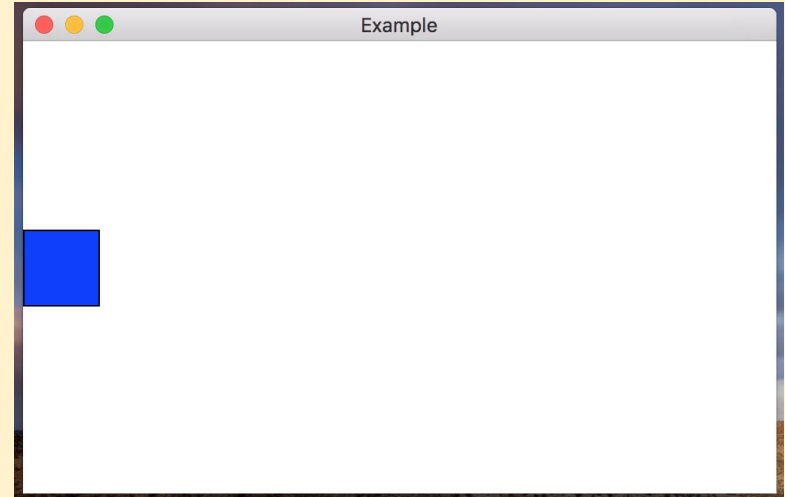
```
def main():  
    gui = graphics(width, height, 'Sup?')  
    while True:  
        gui.clear()  
        # Drawing code here  
        gui.update_frame(30)
```

```
main()
```



Draw This, with a program that has a draw loop

```
def main():  
    gui = graphics(width, height, 'Sup?')  
    while True:  
        gui.clear()  
        # Drawing code here  
        gui.update_frame(30)  
  
main()
```



```
from graphics import graphics
```

```
def main():
```

```
    gui = graphics(500, 300, 'Example')
```

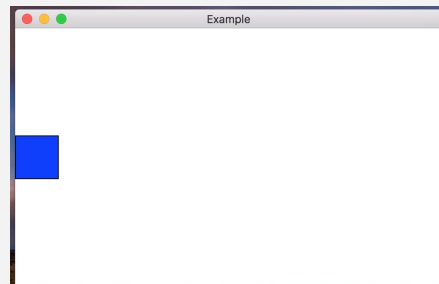
```
    while True:
```

```
        gui.clear()
```

```
        gui.rectangle(0, 125, 50, 50, 'blue')
```

```
        gui.update_frame(30)
```

```
main()
```



Make the rectangle move



Make the rectangle move

```
from graphics import graphics

def main():
    gui = graphics(500, 300, 'Example')
    while True:
        gui.clear()
        gui.rectangle(0, 125, 50, 50, 'blue')
        gui.update_frame(30)

main()
```



```
from graphics import graphics
```

```
def main():
```

```
    gui = graphics(500, 300, 'Example')
```

```
    x_coord = 0
```

```
    while True:
```

```
        gui.clear()
```

```
        gui.rectangle(x_coord, 125, 50, 50, 'blue')
```

```
        gui.update_frame(30)
```

```
        x_coord += 2
```

```
main()
```

Make the
rectangle
move,
wrapping
around



Make the rectangle move, wrapping around

```
def main():  
    gui = graphics(500, 300, 'Example')  
    x_coord = 0  
    while True:  
        gui.clear()  
        gui.rectangle(x_coord, 125, 50, 50, 'blue')  
        gui.update_frame(30)  
        x_coord += 2  
  
main()
```



```
from graphics import graphics
```

```
def main():
```

```
    gui = graphics(500, 300, 'Example')
```

```
    x_coord = -50
```

```
    while True:
```

```
        gui.clear()
```

```
        gui.rectangle(x_coord, 125, 50, 50, 'blue')
```

```
        gui.update_frame(60)
```

```
        x_coord += 10
```

```
        if x_coord > 550:
```

```
            x_coord = -50
```

```
main()
```



Make the rectangle move



Make the rectangle move

```
def main():  
    gui = graphics(500, 300, 'Example')  
    x_coord = 0  
    while True:  
        gui.clear()  
        gui.rectangle(x_coord - 100, 125, 50, 50, 'blue')  
        gui.update_frame(60)  
        x_coord += 10  
        if x_coord > 550:  
            x_coord = -50
```



```
from graphics import graphics
```

```
def main():
```

```
    gui = graphics(500, 300, 'Example')
```

```
    x_coord = 0
```

```
    y_coord = 0
```

```
    while True:
```

```
        gui.clear()
```

```
        gui.rectangle(x_coord, y_coord, 50, 50, 'blue')
```

```
        gui.update_frame(30)
```

```
        x_coord += 5
```

```
        y_coord += 3
```

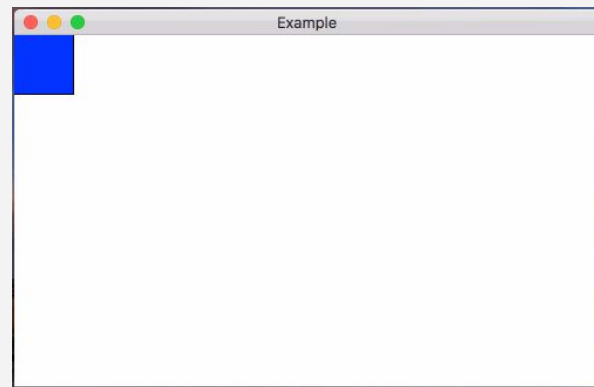
```
main()
```


Mouse Position

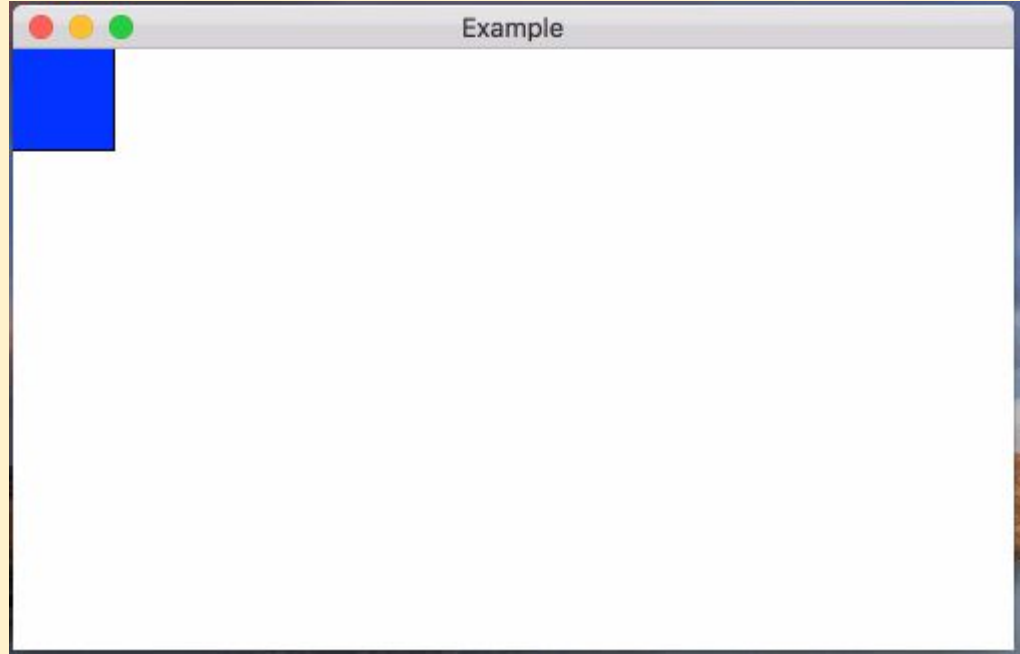
- You can access the x and y coordinates of the mouse via the **graphics** object

`graphics.mouse_x`

`graphics.mouse_y`



Make the
square move
with the
mouse

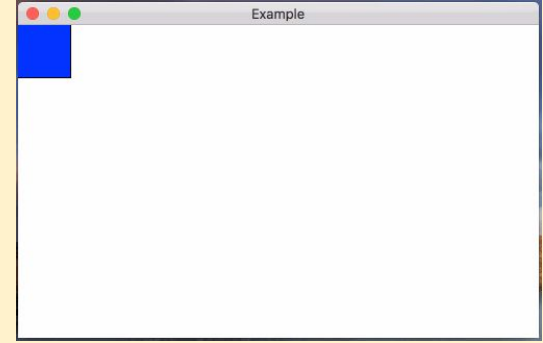


What should the x and y coordinates be?

```
from graphics import graphics
```

```
def main():  
    gui = graphics(500, 300, 'Example')  
    while True:  
        gui.clear()  
        gui.rectangle(???, ???, 100, 100, 'blue')  
        gui.update_frame(60)
```

```
main()
```



```
from graphics import graphics
```

```
def main():
```

```
    gui = graphics(500, 300, 'Example')
```

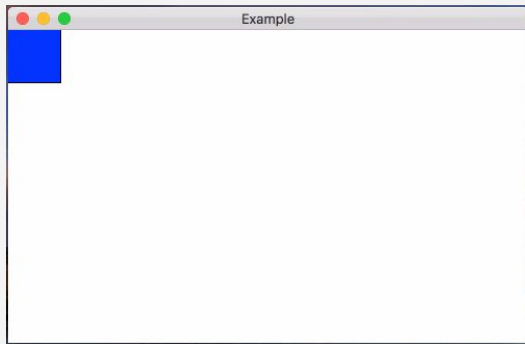
```
    while True:
```

```
        gui.clear()
```

```
        gui.rectangle(gui.mouse_x - 50, gui.mouse_y - 50, 100, 100, 'blue')
```

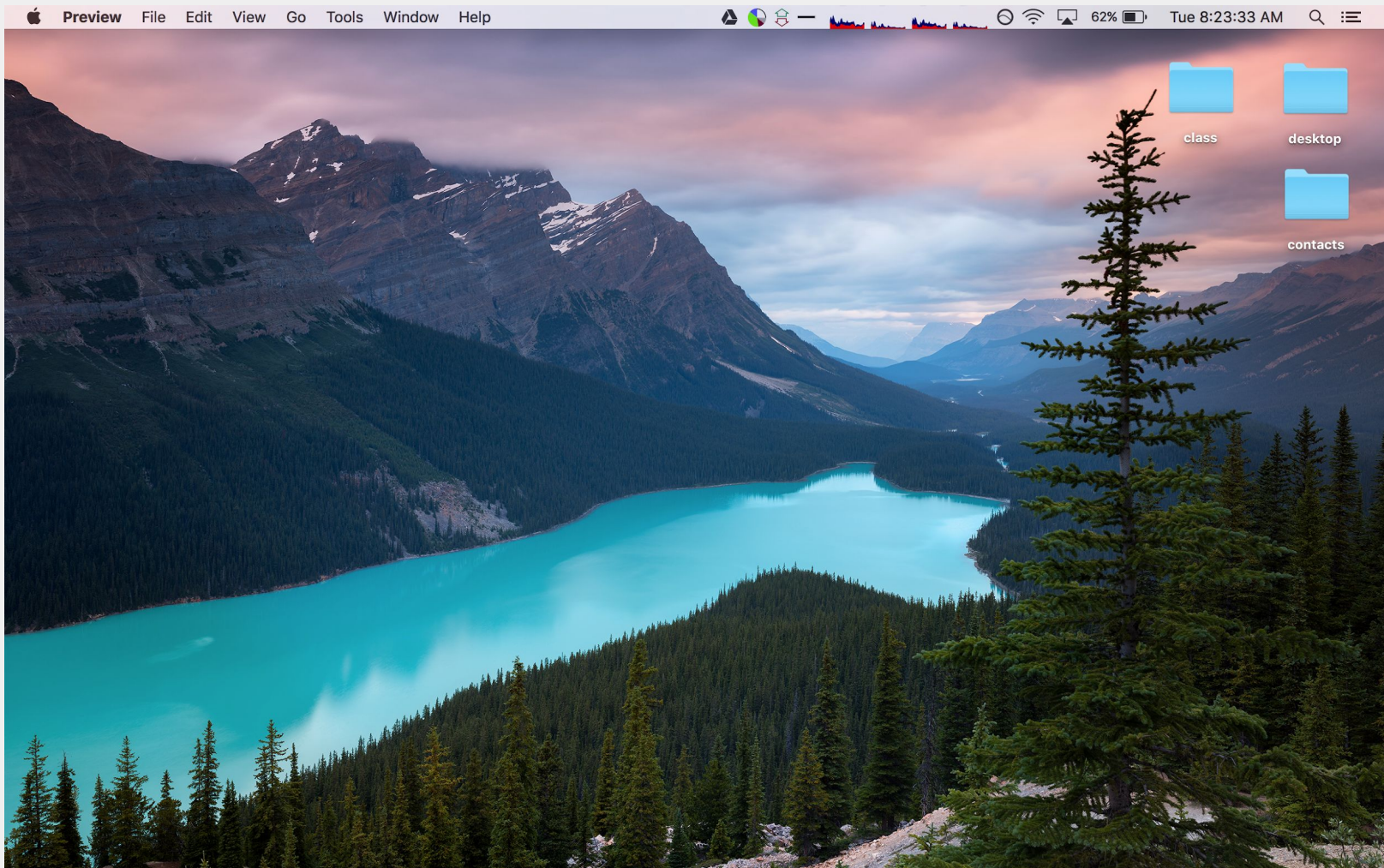
```
        gui.update_frame(60)
```

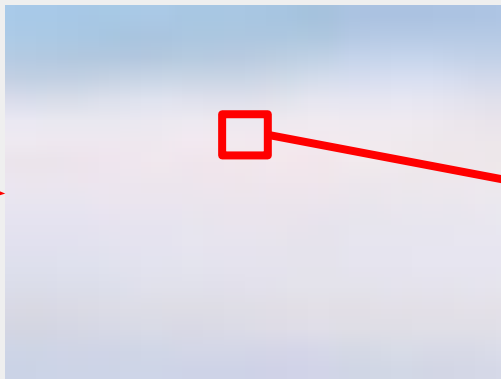
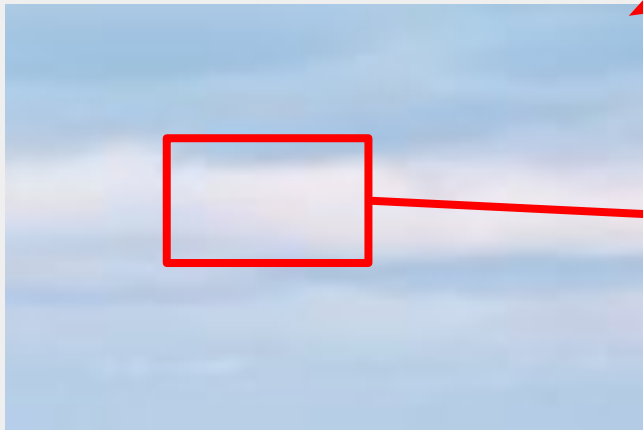
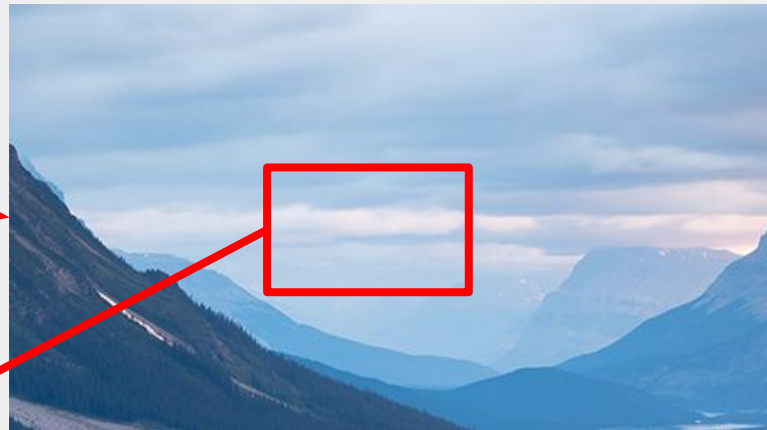
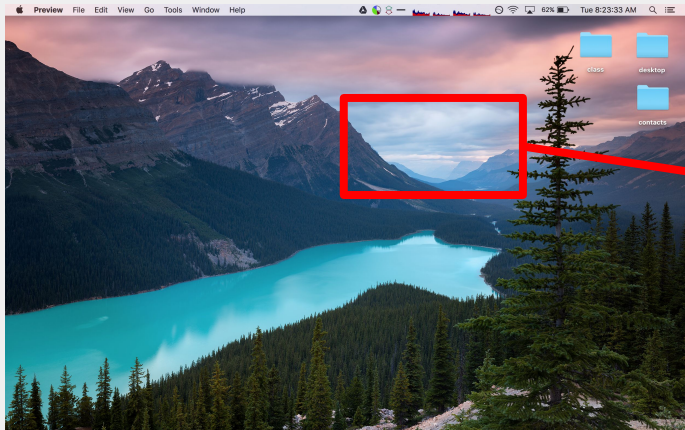
```
main()
```

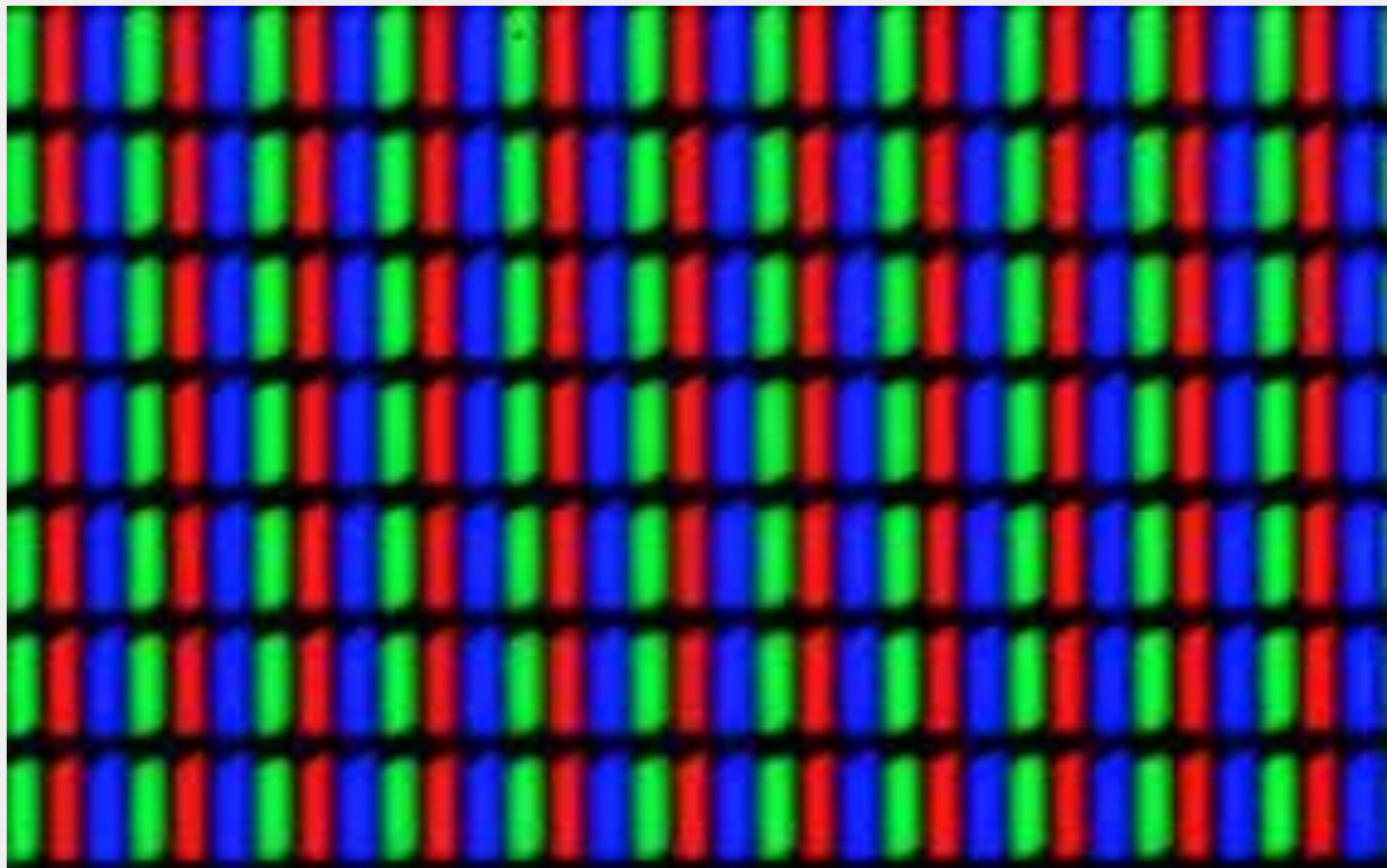


Color

- A computer screen is composed of a grid (rows and columns) of pixels
 - ... Like the canvas of a processing window
- Each pixel can display a unique color, which is controlled by a **Red**, **Green**, and **Blue** brightness value







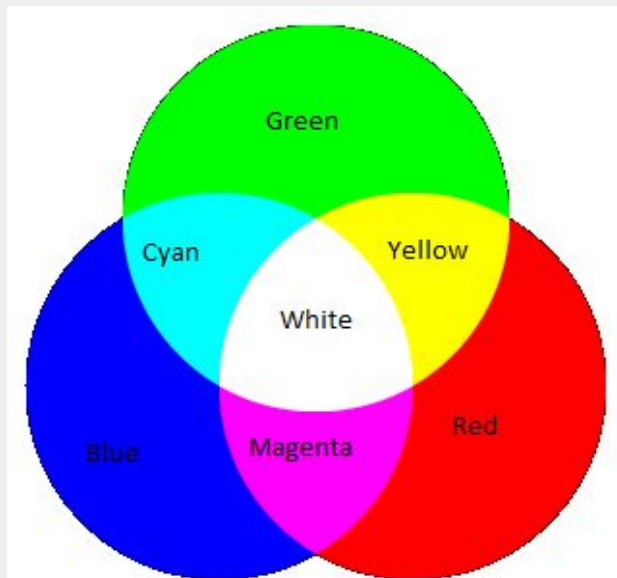
Color

- Thus, when we specify to a computer program, we can do so using these three numbers
- The brightness values for each can be between 0 (the darkest) and 255 (the highest)
 - 0-255 inclusive

Color

- Thus, when we specify to a computer program, we can do so using these three numbers
- The brightness values for each can be between 0 (the darkest) and 255 (the highest)
 - 0-255 inclusive

R G B
(250, 150, 50)



Determine the colors

R G B

- A. What color is (255, 0, 0) ?
- B. What color is (100, 200, 255) ?
- C. What color is (150, 200, 0) ?

Solutions

A. What color is (255, 0, 0) ?



B. What color is (100, 200, 255) ?

C. What color is (150, 200, 0) ?

Solutions

A. What color is (255, 0, 0) ?



B. What color is (100, 200, 255) ?



C. What color is (150, 200, 0) ?

Solutions

A. What color is (255, 0, 0) ?



B. What color is (100, 200, 255) ?



C. What color is (150, 200, 0) ?



What might the RGB values be?

A. Match this color:



B. Match this color:



C. Match this color:



What might the RGB values be?

A. Match this color: 118, 253, 138

B. Match this color:



C. Match this color:

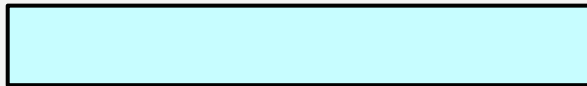


What might the RGB values be?

A. Match this color: 118, 253, 138

B. Match this color: 255, 251, 55

C. Match this color:



What might the RGB values be?

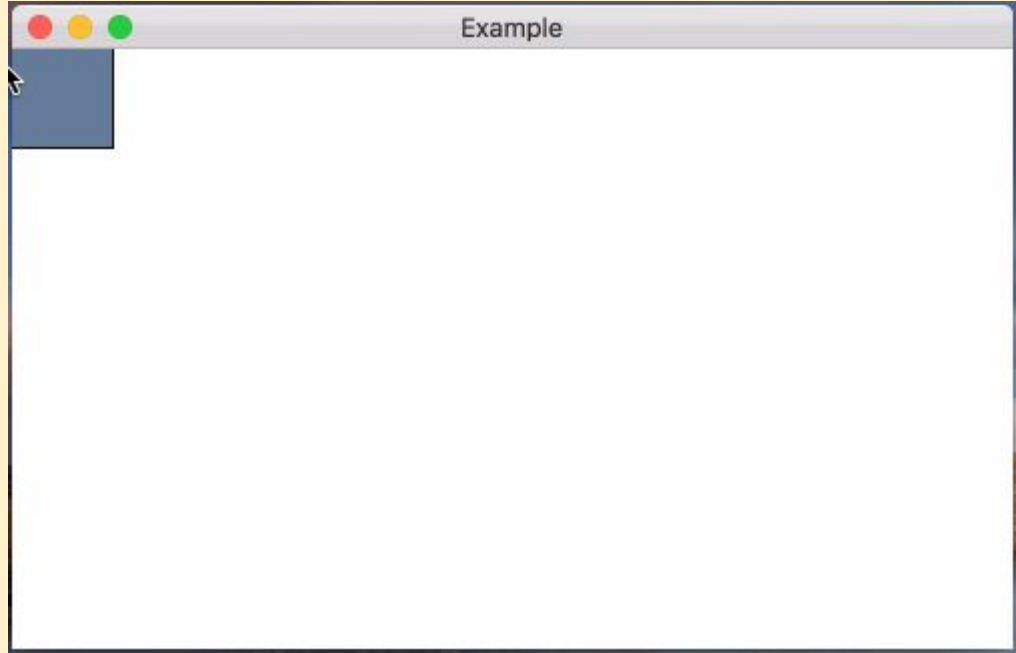
- A. Match this color: 118, 253, 138
- B. Match this color: 255, 251, 55
- C. Match this color: 201, 253, 253

Getting a color string from RGB values

- The argument passed as the **fill** to the shape functions is expected to be a string
- How can we specify a color with RGB then?

```
color_string = gui.get_color_string(100, 150, 200)
gui.rectangle(50, 50, 50, 50, color_string)
```

Make the color
change
roughly each
second



Start from here!

```
from graphics import graphics
```

```
def main():
```

```
    gui = graphics(500, 300, 'Example')
```

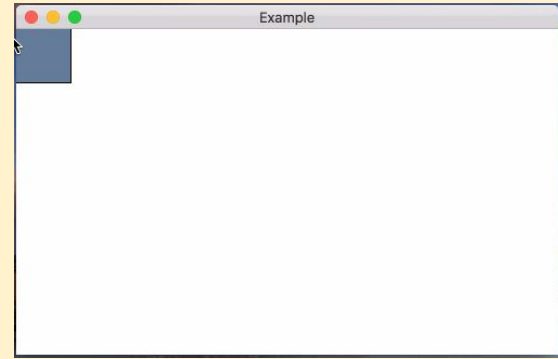
```
    while True:
```

```
        gui.clear()
```

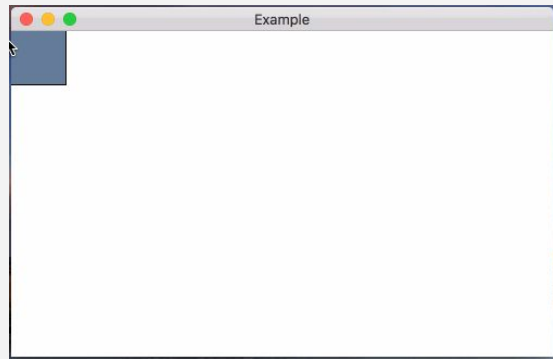
```
        gui.rectangle(gui.mouse_x - 50, gui.mouse_y - 50, 100, 100, 'blue')
```

```
        gui.update_frame(60)
```

```
main()
```



```
gui = graphics(500, 300, 'Example')
color_string = 'blue'
i = 0
while True:
    if i % 60 == 0:
        red = random.randint(0, 255)
        green = random.randint(0, 255)
        blue = random.randint(0, 255)
        color_string = gui.get_color_string(red, green, blue)
    gui.clear()
    gui.rectangle(gui.mouse_x - 50, gui.mouse_y - 50, 100, 100, color_string)
    gui.update_frame(60)
    i += 1
```



http://www.science.smith.edu/dftwiki/index.php/Color_Charts_for_TKinter

Named colour chart												
snow	deep sky blue	gold	seashell3	SlateBlue2	LightBlue3	SpringGreen2	DarkGoldenrod1	brown4	pink3	purple1	gray26	gray64
ghost white	sky blue	light goldenrod	seashell4	SlateBlue3	LightBlue4	SpringGreen3	DarkGoldenrod2	salmon1	pink4	purple2	gray27	gray65
white smoke	light sky blue	goldenrod	AntiqueWhite1	SlateBlue4	LightCyan2	SpringGreen4	DarkGoldenrod3	salmon2	LightPink1	purple3	gray28	gray66
gainsboro	steel blue	dark goldenrod	AntiqueWhite2	RoyalBlue1	LightCyan3	green2	DarkGoldenrod4	salmon3	LightPink2	purple4	gray29	gray67
floral white	light steel blue	rosy brown	AntiqueWhite3	RoyalBlue2	LightCyan4	green3	RosyBrown1	salmon4	LightPink3	MediumPurple1	gray30	gray68
old lace	light blue	indian red	AntiqueWhite4	RoyalBlue3	PaleTurquoise1	green4	RosyBrown2	LightSalmon2	LightPink4	MediumPurple2	gray31	gray69
linen	powder blue	saddle brown	bisque2	RoyalBlue4	PaleTurquoise2	chartreuse2	RosyBrown3	LightSalmon3	PaleVioletRed1	MediumPurple3	gray32	gray70
antique white	pale turquoise	sandy brown	bisque3	blue2	PaleTurquoise3	chartreuse3	RosyBrown4	LightSalmon4	PaleVioletRed2	MediumPurple4	gray33	gray71
papaya whip	dark turquoise	dark salmon	bisque4	blue4	PaleTurquoise4	chartreuse4	IndianRed1	orange2	PaleVioletRed3	thistle1	gray34	gray72
blanched almond	medium turquoise	salmon	PeachPuff2	DodgerBlue2	CadetBlue1	OliveDrab1	IndianRed2	orange3	PaleVioletRed4	thistle2	gray35	gray73
bisque	turquoise	light salmon	PeachPuff3	DodgerBlue3	CadetBlue2	OliveDrab2	IndianRed3	orange4	maroon1	thistle3	gray36	gray74
peach puff	cyan	orange	PeachPuff4	DodgerBlue4	CadetBlue3	OliveDrab4	IndianRed4	DarkOrange1	maroon2	thistle4	gray37	gray75
navajo white	light cyan	dark orange	NavajoWhite2	SteelBlue1	CadetBlue4	DarkOliveGreen1	sienna1	DarkOrange2	maroon3		gray38	gray76
lemon chiffon	cadet blue	coral	NavajoWhite3	SteelBlue2	turquoise1	DarkOliveGreen2	sienna2	DarkOrange3	maroon4		gray39	gray77
mint cream	medium aquamarine	light coral	NavajoWhite4	SteelBlue3	turquoise2	DarkOliveGreen3	sienna3	DarkOrange4	VioletRed1		gray40	gray78
azure	aquamarine	tomato	LemonChiffon2	SteelBlue4	turquoise3	DarkOliveGreen4	sienna4	coral1	VioletRed2		gray41	gray79
alice blue	dark green	orange red	LemonChiffon3	DeepSkyBlue2	turquoise4	khaki1	burlywood1	coral2	VioletRed3		gray42	gray80
lavender	dark olive green	red	LemonChiffon4	DeepSkyBlue3	cyan2	khaki2	burlywood2	coral3	VioletRed4		gray43	gray81
lavender blush	dark sea green	hot pink	cornsilk2	DeepSkyBlue4	cyan3	khaki3	burlywood3	coral4	magenta2		gray44	gray82
misty rose	sea green	deep pink	cornsilk3	SkyBlue1	cyan4	khaki4	burlywood4	tomato2	magenta3		gray45	gray83
dark slate gray	medium sea green	pink	cornsilk4	SkyBlue2	DarkSlateGray1	LightGoldenrod1	wheat1	tomato3	magenta4		gray46	gray84
dim gray	light sea green	light pink	ivory2	SkyBlue3	DarkSlateGray2	LightGoldenrod2	wheat2	tomato4	orchid1		gray47	gray85
slate gray	pale green	pale violet red	ivory3	SkyBlue4	DarkSlateGray3	LightGoldenrod3	wheat3	OrangeRed2	orchid2		gray48	gray86
light slate gray	spring green	maroon	ivory4	LightSkyBlue1	DarkSlateGray4	LightGoldenrod4	wheat4	OrangeRed3	orchid3		gray49	gray87
gray	lawn green	medium violet red	honeydew2	LightSkyBlue2	aquamarine2	LightYellow2	tan1	OrangeRed4	orchid4		gray50	gray88
light gray	medium spring green	violet red	honeydew3	LightSkyBlue3	aquamarine4	LightYellow3	tan2	red2	plum1		gray51	gray89
midnight blue	green yellow	medium orchid	honeydew4	LightSkyBlue4	DarkSeaGreen1	LightYellow4	tan4	red3	plum2		gray52	gray90
navy	lime green	dark orchid	LavenderBlush2	SlateGray1	DarkSeaGreen2	yellow2	chocolate1	red4	plum3		gray53	gray91
cornflower blue	yellow green	dark violet	LavenderBlush3	SlateGray2	DarkSeaGreen3	yellow3	chocolate2	DeepPink2	plum4		gray54	gray92
dark slate blue	forest green	blue violet	LavenderBlush4	SlateGray3	DarkSeaGreen4	yellow4	chocolate3	DeepPink3	MediumOrchid1		gray55	gray93
slate blue	olive drab	purple	MistyRose2	SlateGray4	SeaGreen1	gold2	firebrick1	DeepPink4	MediumOrchid2		gray56	gray94
medium slate blue	dark khaki	medium purple	MistyRose3	LightSteelBlue1	SeaGreen2	gold3	firebrick2	HotPink1	MediumOrchid3		gray57	gray95
light slate blue	khaki	thistle	MistyRose4	LightSteelBlue2	SeaGreen3	gold4	firebrick3	HotPink2	MediumOrchid4		gray58	gray96
medium blue	pale goldenrod	snow2	azure2	LightSteelBlue3	PaleGreen1	goldenrod1	firebrick4	HotPink3	DarkOrchid1		gray59	gray97
royal blue	light goldenrod yellow	snow3	azure3	LightSteelBlue4	PaleGreen2	goldenrod2	brown1	HotPink4	DarkOrchid2		gray60	gray98
blue	light yellow	snow4	azure4	LightBlue1	PaleGreen3	goldenrod3	brown2	pink1	DarkOrchid3		gray61	gray99
dodger blue	yellow	seashell2	SlateBlue1	LightBlue2	PaleGreen4	goldenrod4	brown3	pink2	DarkOrchid4		gray62	gray63