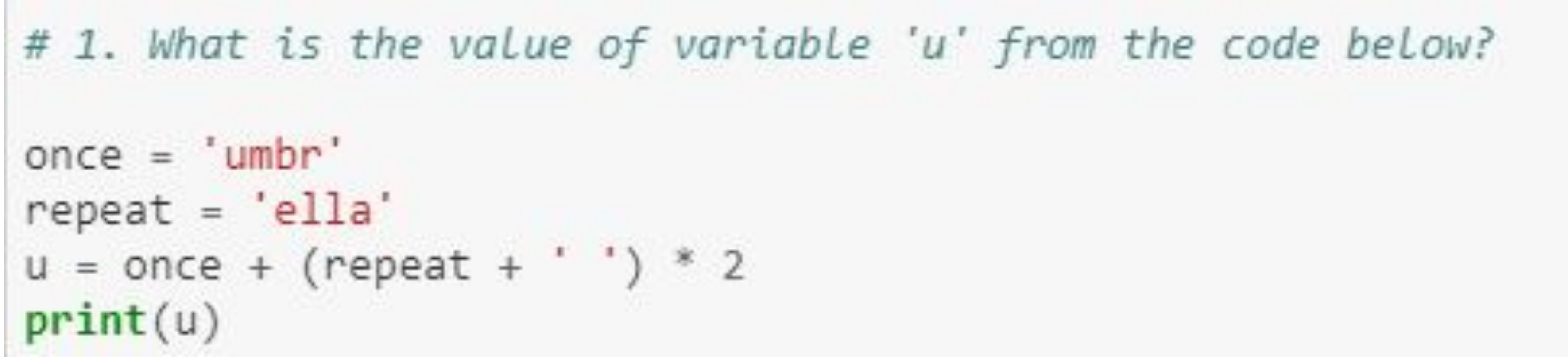
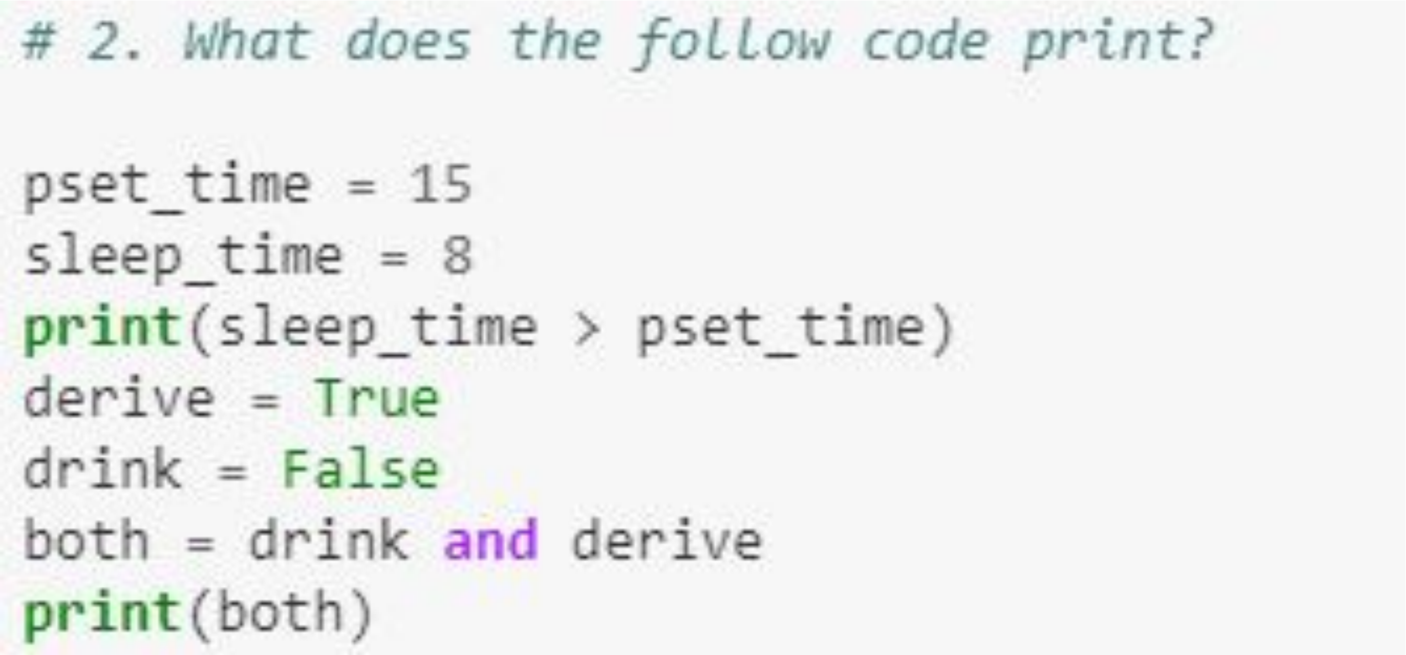
Drawing

Making Game with Python

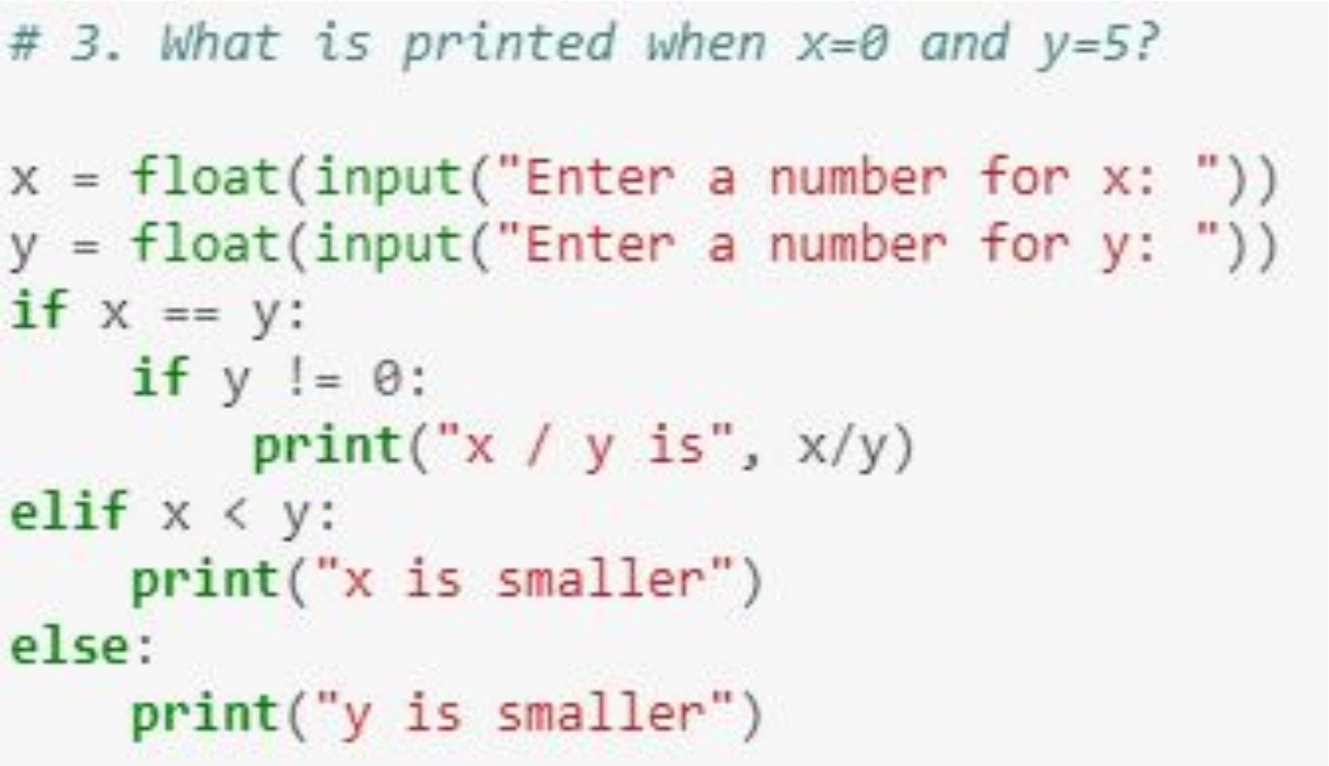
Exercise



```
# 1. What is the value of variable 'u' from the code below?

once = 'umbr'
repeat = 'ella'
u = once + (repeat + ' ') * 2
print(u)
```



```
# 2. What does the follow code print?

pset_time = 15
sleep_time = 8
print(sleep_time > pset_time)
derive = True
drink = False
both = drink and derive
print(both)
```



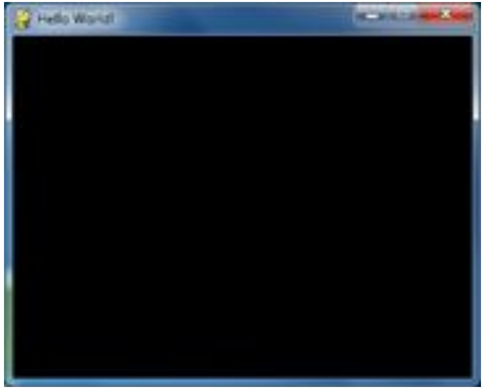
```python
# 3. What is printed when x=0 and y=5?

x = float(input("Enter a number for x: "))
y = float(input("Enter a number for y: "))
if x == y:
    if y != 0:
        print("x / y is", x/y)
elif x < y:
    print("x is smaller")
else:
    print("y is smaller")
```

Last time

- Pygame basics
  - Surface object
  - Rect object
  - Drawing images
- Bouncing ball game

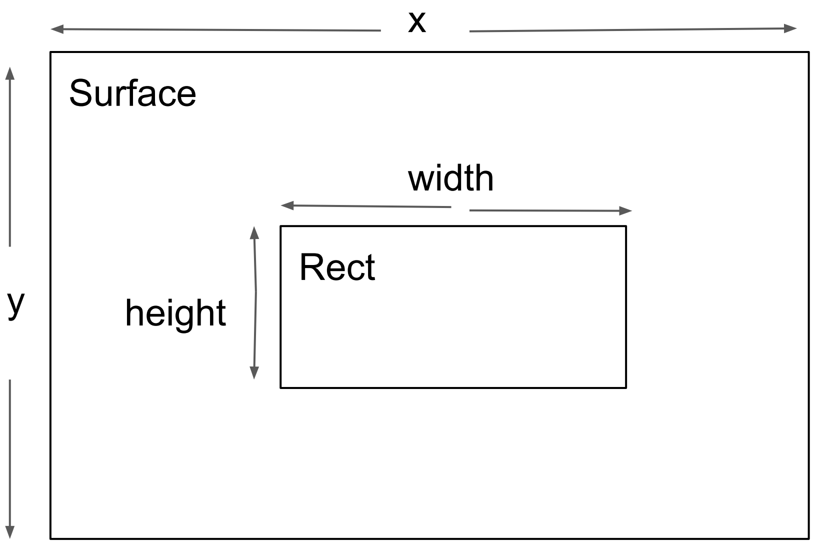
Surface Object

- Surface objects represent rectangle 2D images
  - Display surface: pygame.display.set_mode((width, height))
  - pygame.display.set_caption(‘Hello World’)



Rect Object

- Tuple of 4 integers
  - X coordinate of the top left corner (pixel)
  - Y coordinate of the top left corner (pixel)
  - Width of the rectangle (pixel)
  - Height of the rectangle (pixel)
- Create Rect Object
  - pygame.Rect(10, 20, 200, 300)



Today

- Pygame basics
    - Color
    - Primitive drawing function
- Drawing program

Colors

- Three primary colors
  - Red, Green, Blue (RGB)

- Other colors are the combination of different amount of RGB

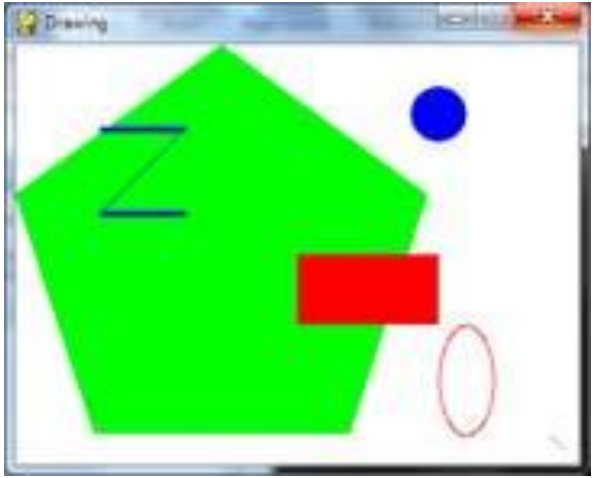
Colors

- Pygame Color objects
  - Tuples of three integers (0-255) representing (R,G,B)

Color	Tuple
Red	(255, 0, 0)
Green	(0, 255, 0)
Blue	(0, 0, 255)
White	(255, 255, 255)
Black	(0, 0, 0)
Yellow	(255, 255, 0)

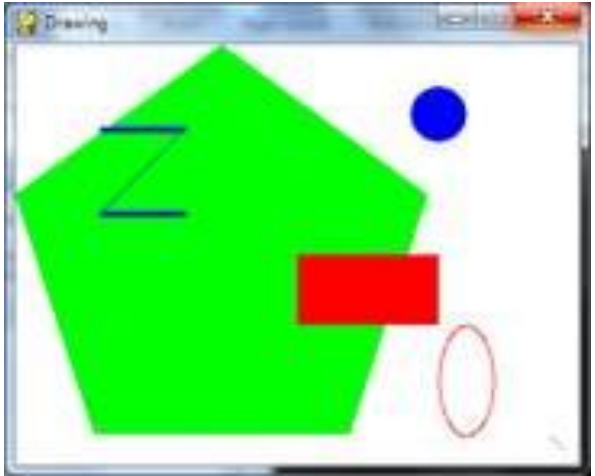
Primitive Drawing Functions

- Draw rectangle
  - pygame.draw.rect(surface, color, rectangle, width=0)
- Drawing polygon
  - pygame.draw.polygon(surface, color, vertices, width=0)



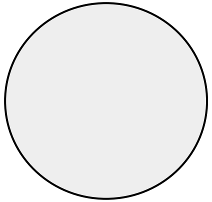
Primitive Drawing Functions (2)

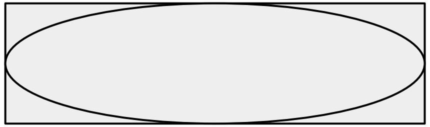
- Drawing line
  - pygame.draw.line(surface, color, start_point, end_point, width=1)
- Drawing lines
  - pygame.draw.lines(surface, color, closed, vertices, width=1)



Primitive Drawing Functions (3)

- Drawing a circle
  - pygame.draw.circle(surface, color, center, radius, width=1)
- Drawing ellipse
  - pygame.draw.ellipse(surface, color, bounding_rectangle, width=1)





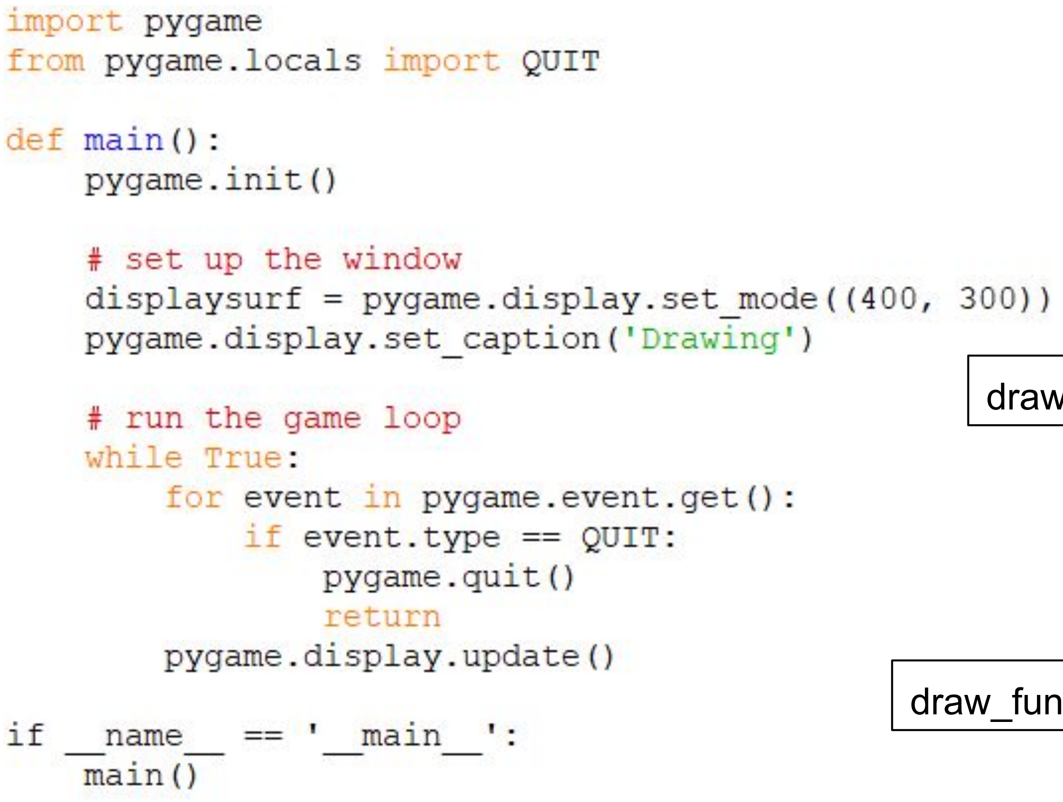
Primitive drawing functions (4)

- PixelArray object
  - pixelObj = pygame.PixelArray(surface)
  - pixelObj[x][y] = color

Primitive drawing functions (4)

- PixelArray object
  - pixelObj = pygame.PixelArray(surface)
  - pixelObj[x][y] = color

Putting together: base case

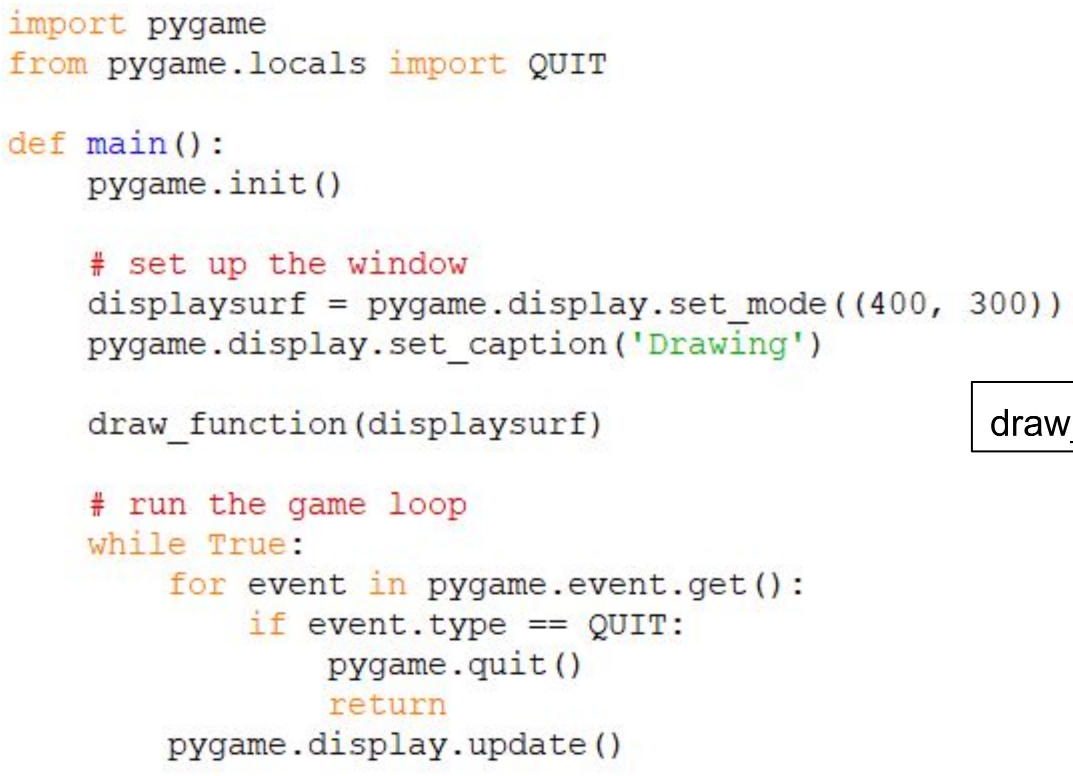


```
import pygame
from pygame.locals import QUIT

def main():
    pygame.init()

    # set up the window
    displaysurf = pygame.display.set_mode((400, 300))
    pygame.display.set_caption('Drawing')

    # run the game loop
    while True:
        for event in pygame.event.get():
            if event.type == QUIT:
                pygame.quit()
                return
        pygame.display.update()

if __name__ == '__main__':
    main()
```

draw_function call will be added here

draw_function definition will be added here

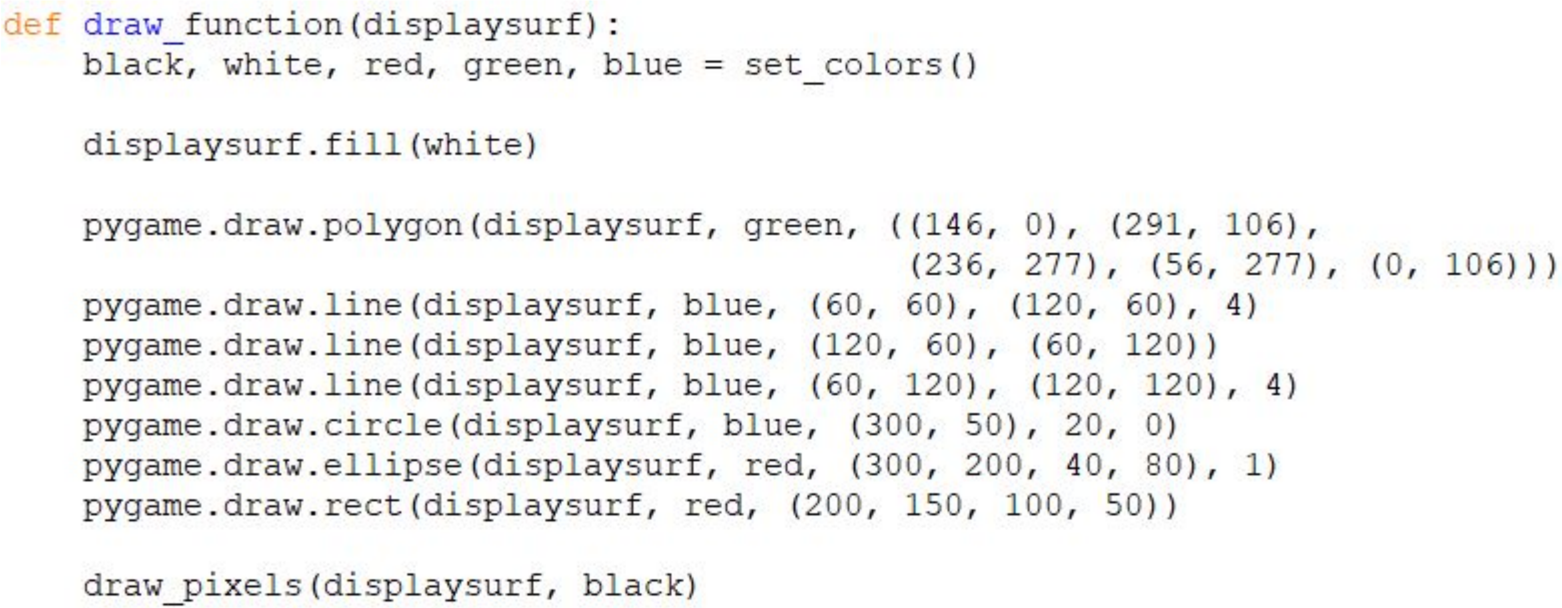
Putting together: drawing.py (1)



```
import pygame
from pygame.locals import QUIT

def main():
    pygame.init()

    # set up the window
    displaysurf = pygame.display.set_mode((400, 300))
    pygame.display.set_caption('Drawing')

    draw_function(displaysurf)

    # run the game loop
    while True:
        for event in pygame.event.get():
            if event.type == QUIT:
                pygame.quit()
                return
        pygame.display.update()
```

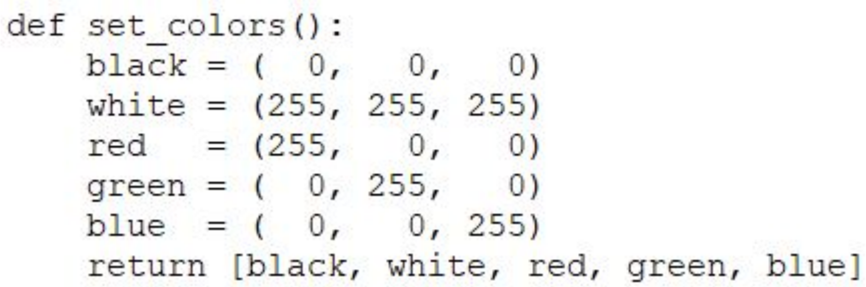
draw_function is called here

Putting together: drawing.py (2)



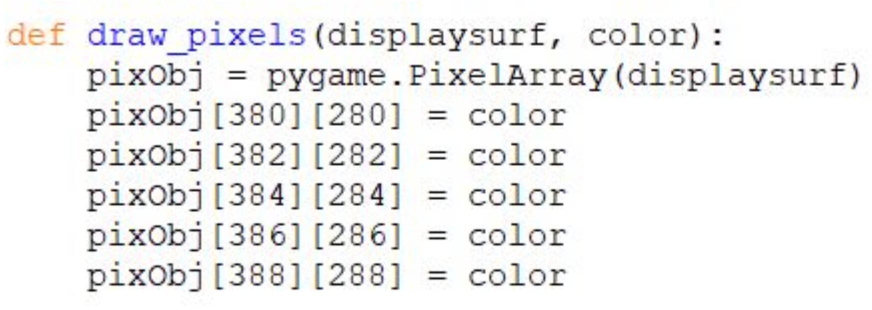
```
def draw_function(displaysurf):
    black, white, red, green, blue = set_colors()

    displaysurf.fill(white)

    pygame.draw.polygon(displaysurf, green, ((146, 0), (291, 106),
                                             (236, 277), (56, 277), (0, 106)))
    pygame.draw.line(displaysurf, blue, (60, 60), (120, 60), 4)
    pygame.draw.line(displaysurf, blue, (120, 60), (60, 120))
    pygame.draw.line(displaysurf, blue, (60, 120), (120, 120), 4)
    pygame.draw.circle(displaysurf, blue, (300, 50), 20, 0)
    pygame.draw.ellipse(displaysurf, red, (300, 200, 40, 80), 1)
    pygame.draw.rect(displaysurf, red, (200, 150, 100, 50))

    draw_pixels(displaysurf, black)
```

Putting together: drawing.py (3)



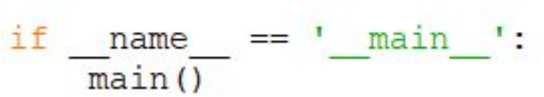
```python
def set_colors():
    black = (  0,   0,   0)
    white = (255, 255, 255)
    red   = (255,   0,   0)
    green = (  0, 255,   0)
    blue  = (  0,   0, 255)
    return [black, white, red, green, blue]
```

Putting together: drawing.py (4)



```
def draw_pixels(displaysurf, color):
    pixObj = pygame.PixelArray(displaysurf)
    pixObj[380][280] = color
    pixObj[382][282] = color
    pixObj[384][284] = color
    pixObj[386][286] = color
    pixObj[388][288] = color
```

Putting together: drawing.py (4)



```
if __name__ == '__main__':
    main()
```

Q&A

https://github.com/zhihongzeng2002/pythongame/tree/master/2