# Pygame installation

- Before installing Pygame, Python should be installed in the system, and it is good to have 3.6.1 or above version because it is much friendlier to beginners, and additionally runs faster
- **Installing through pip:** The good way to install Pygame is with the pip tool (which is what python uses to install packages). The command is the following:

**Pip install pygame**

# Technical activity

## Step 1- import required modules

import sys and pygame module then initialize the pygame using init() method.

```
pygame.init()
```

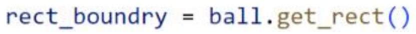
## Step 2- set screen size, speed of ball, background colour and caption.

The pygame.display.set_mode() function returns the surface object for the window. This function accepts width and height of the screen as arguments. To set the caption of the window, call pygame.display.set_caption() function.

```
pygame.display.set_caption('Bouncing ball')
```

## Step 3- load the ball image and set the rectangle area covering image

Here, we have opened the image using pygame.image.load() method and set the ball rectangle area boundary using get_rect() method.



```
rect_boundry = ball.get_rect()
```

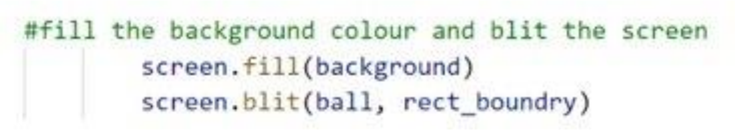
## Step 4-make ball movement continuity

We have created an infinite loop to make the ball move continuously and reverse the direction of ball if it hits the edges.

```
            speed[1] = -speed[1]
```

### Step 5-fill the background colour and blit the screen

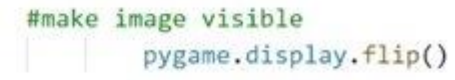
Here, fill() method is used to fill the surface background color. In step 3, we have mentioned how to create the moving object. But, we also need to render them on the window. In pygame, this is called blitting of an object and implemented with blit() method.



```
#fill the background colour and blit the screen
        screen.fill(background)
        screen.blit(ball, rect_boundry)
```

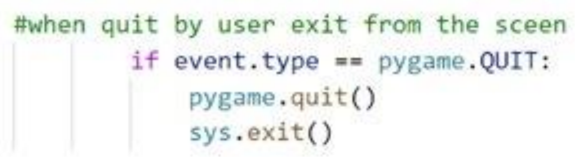
### Step 6-fill the background colour and blit the screen

We have used the pygame flip() method to make all image visible.



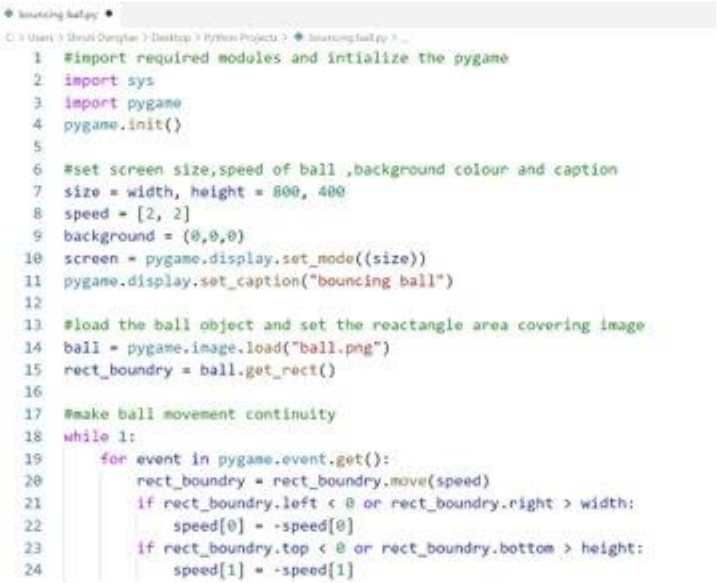
```
#make image visible
        pygame.display.flip()
```

### Step 7-exit from the screen

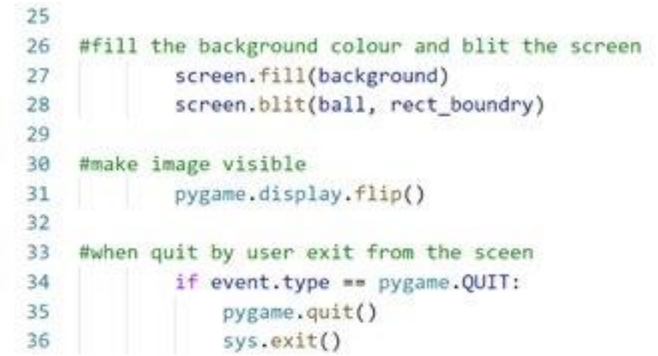


```
#when quit by user exit from the sceen
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
```

## Final code



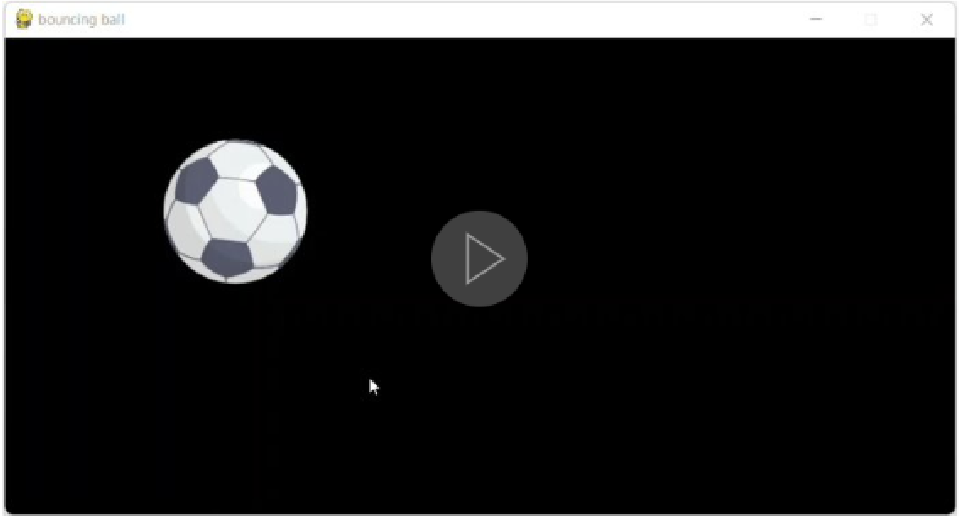
```python
#import required modules and intialize the pygame
import sys
import pygame
pygame.init()

#set screen size,speed of ball ,background colour and caption
size = width, height = 800, 400
speed = [2, 2]
background = (0,0,0)
screen = pygame.display.set_mode((size))
pygame.display.set_caption("bouncing ball")

#load the ball object and set the reactangle area covering image
ball = pygame.image.load("ball.png")
rect_boundry = ball.get_rect()

#make ball movement continuity
while 1:
    for event in pygame.event.get():
        rect_boundry = rect_boundry.move(speed)
        if rect_boundry.left < 0 or rect_boundry.right > width:
            speed[0] = -speed[0]
        if rect_boundry.top < 0 or rect_boundry.bottom > height:
            speed[1] = -speed[1]
```



```python

#fill the background colour and blit the screen
        screen.fill(background)
        screen.blit(ball, rect_boundry)

#make image visible
        pygame.display.flip()

#when quit by user exit from the sceen
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
```

# Final image output-

run the program by ctrl+f5



# Final video output-

refer slide 11