

```
import pygame

## initialize pygame
pygame.init()

## create variables that store the rgb values for colors
## so we can use it later without writing the rgb numbers each time
## you can pick any colors you want
black = (0,0,0)
green = (35, 232, 61)
grey = (9,105,16)
orange = (255,187,0)
white = (255,255,255)

## screen size
## you may chose to make your screen bigger or smaller
## width, height
size = 700, 500

## initialize a the screen to be displayed
screen = pygame.display.set_mode(size)

## window title
pygame.display.set_caption("Pygame Tutorial")

## will become true when we close the screen
game_over = False

## position of box
## the box will serve as an obstacle later on
## you can position the box anywhere on the screen
box_x = 300
box_y = 410

## size of box
## our box will be a square, so we only have one size used for the height and the width
box_size = 40

## player will also be shown as a box, but you will be able to move it right and left
## these are the starting coordinates
player_x = 50
player_y = 420

## also a square with one size for height and width
player_size = 30

## initialize the speed to 0, because we don't want the box to move on its own
player_speed_x = 0
```

```

## we create a clock to help us keep track of time
clock = pygame.time.Clock()

## parameters for pygame.draw.rect(screen, color, [x_pos, y_pos, height, width])
## displays the player square on the screen, when called
def display_player(p_x, p_y):
    pygame.draw.rect(screen, green, [p_x, p_y, player_size, player_size])

def display_box():
    pygame.draw.rect(screen, orange, [box_x, box_y, box_size, box_size])

## draws a rectangle at the bottom of the screen that serves as a ground, optional
def display_ground():
    pygame.draw.rect(screen, grey, [0, 450, 700, 50])

## game loop: runs until game_over is True
while not game_over:
    ## event = any inputs given, ex: a key being pressed or a mouse click
    for event in pygame.event.get():

        ## if we click on quit button (red X at the top right corner)
        if event.type == pygame.QUIT:
            ## then game is over
            game_over = True

        ## if the event is any key being presses on the keyboard
        if event.type == pygame.KEYDOWN:

            ## if the key pressed is LEFT ARROW KEY
            if event.key == pygame.K_LEFT:
                ## move to the left by setting the speed to -10 and adding that to the current
                ## x position of the player
                player_speed_x = -10
                player_x += player_speed_x

            ## if the key pressed is RIGHT ARROW KEY
            if event.key == pygame.K_RIGHT:
                ## move to the right by increasing the position by 10
                player_speed_x = 10
                player_x += player_speed_x

    ## fills the background of the screen with black
    screen.fill(black)
    display_ground()
    display_player(player_x, player_y)
    display_box()

```

```
## update the contents of the entire display  
pygame.display.flip()
```

```
## should be called every loop, will determine how much time has passed since the  
## the previous call  
## the argument is optional - limits the runtime speed of the game  
## in out case, the program never runs at more than 40 frames per second  
clock.tick(60)
```

```
## quit pygame  
pygame.quit()
```