

Code Library · Stage 3

Six tiny programs that play with COLOUR. Copy, paste, and tweak.

First, a tiny bit of background

On a computer screen, every colour is just three numbers - (**red, green, blue**), each between 0 and 255. **0** means that light is OFF, **255** means full brightness. Mix all three and you get every colour you can see.

Quick examples:

- (0, 0, 0) - everything off → **black**
- (255, 255, 255) - everything on → **white**
- (255, 0, 0) - only red → **pure red**
- (0, 255, 0) - only green → **pure green**
- (255, 215, 0) - red plus a lot of green → **gold/yellow**

Typing (255, 215, 0) over and over gets old fast. Instead, save each colour into a **variable** with a clear ALL_CAPS name and use the name everywhere. It makes your code read like a sentence.

```
# the trick of Stage 3
YELLOW = (255, 215, 0)
SPACE  = (0, 0, 50)

screen.fill(SPACE)
pygame.draw.rect(screen, YELLOW, (350, 250, 100, 100))
```

How to use this handout

Each game has three parts: **what it does**, **the code**, and **challenges**. Copy any code block into the Trinket editor on the lesson page, then press the green  **Run** button. Change a number, a colour, a word - see what happens.

Game 1 · Palette Test Card

What it does

Defines a palette of named colours and paints six rectangles side by side - one per colour. Like swatches at a paint store.

```
import pygame

pygame.init()
screen = pygame.display.set_mode((600, 200))
pygame.display.set_caption("Palette Test Card")

RED    = (220, 60, 60)
ORANGE = (240, 150, 60)
YELLOW = (240, 210, 70)
GREEN  = (90, 180, 100)
BLUE   = (70, 130, 220)
PURPLE = (160, 80, 200)

palette = [RED, ORANGE, YELLOW, GREEN, BLUE, PURPLE]

running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
    for i, colour in enumerate(palette):
        pygame.draw.rect(screen, colour, (i * 100, 0, 100, 200))
    pygame.display.flip()

pygame.quit()
```

Try these one by one

Palette 1 Add PINK = (255, 105, 180) and TEAL = (0, 180, 180) to the palette.

Palette 2 Replace BLUE with a colour you find on Google's colour picker.

Palette 3 Make the window 800 wide so two more colours fit nicely.

```
screen = pygame.display.set_mode((800, 200))
```

Game 2 · Rainbow Stripes

What it does

Stacks seven horizontal rainbow stripes top to bottom. Same trick as Palette Test Card but turned 90 degrees.

```
import pygame

pygame.init()
screen = pygame.display.set_mode((600, 420))
pygame.display.set_caption("Rainbow Stripes")

RED    = (228, 30, 30)
ORANGE = (240, 130, 30)
YELLOW = (240, 220, 50)
GREEN  = (60, 200, 90)
BLUE   = (50, 130, 230)
INDIGO = (75, 0, 130)
VIOLET = (140, 60, 200)

rainbow = [RED, ORANGE, YELLOW, GREEN, BLUE, INDIGO, VIOLET]

running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
    for i, colour in enumerate(rainbow):
        pygame.draw.rect(screen, colour, (0, i * 60, 600, 60))
    pygame.display.flip()

pygame.quit()
```

Try these one by one

Rainbow 1 Flip the rainbow upside down with `reversed()` in the loop.

```
for i, colour in enumerate(reversed(rainbow)):
```

Rainbow 2 Make the stripes vertical instead of horizontal.

```
pygame.draw.rect(screen, colour, (i * 85, 0, 85, 420))
```

Rainbow 3 Replace VIOLET with hot pink (255, 105, 180).

```
VIOLET = (255, 105, 180)
```

Game 3 · Traffic Light

What it does

Three coloured circles stacked like a traffic light. Press SPACEBAR to switch which one is glowing (the other two go dim).

```
import pygame

pygame.init()
screen = pygame.display.set_mode((300, 600))
pygame.display.set_caption("Traffic Light")

BG      = (40, 40, 50)
DIM      = (60, 60, 70)
RED      = (230, 50, 50)
YELLOW   = (240, 210, 70)
GREEN    = (70, 200, 100)

glowing = 0 # 0=red, 1=yellow, 2=green

running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE:
            glowing = (glowing + 1) % 3

    screen.fill(BG)
    red_c    = RED    if glowing == 0 else DIM
    yellow_c = YELLOW if glowing == 1 else DIM
    green_c  = GREEN  if glowing == 2 else DIM

    pygame.draw.circle(screen, red_c,    (150, 150), 80)
    pygame.draw.circle(screen, yellow_c, (150, 300), 80)
    pygame.draw.circle(screen, green_c,  (150, 450), 80)
    pygame.display.flip()

pygame.quit()
```

Try these one by one

- Traffic 1** Make the dim colour a darker grey - change DIM to (20, 20, 25).
- Traffic 2** Make the lights tiny by changing radius 80 to 30.
- Traffic 3** Add a 4th BLUE light below the green one.

Game 4 · RGB Mixer

What it does

Three little bars, one big swatch. Press R / G / B to add 20 to that channel. Watch the swatch change colour live. The best way to FEEL how RGB works.

```
import pygame

pygame.init()
screen = pygame.display.set_mode((600, 400))
pygame.display.set_caption("RGB Mixer - press R, G, or B")

r, g, b = 100, 100, 100

running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_r: r = (r + 20) % 260
            if event.key == pygame.K_g: g = (g + 20) % 260
            if event.key == pygame.K_b: b = (b + 20) % 260

    screen.fill((30, 30, 40))
    pygame.draw.rect(screen, (r, g, b), (200, 100, 200, 200)) # big swatch
    pygame.draw.rect(screen, (255, 0, 0), (50, 100, 20, r)) # R bar
    pygame.draw.rect(screen, (0, 255, 0), (90, 100, 20, g)) # G bar
    pygame.draw.rect(screen, (0, 0, 255), (130, 100, 20, b)) # B bar
    pygame.display.flip()

pygame.quit()
```

Try these one by one

Mixer 1 Press R only - what colour does the swatch become at the highest value?

Mixer 2 Press R and G the same number of times - what colour do you get?

Mixer 3 Make the step size 30 instead of 20.

```
if event.key == pygame.K_r: r = (r + 30) % 260
```

Game 5 · Random Sky (random + colours)

What it does

Mixes Stage 1 (random) with Stage 3 (colours). Press SPACE and the background becomes a random colour. Every press, a new sky.

```
import pygame
import random

pygame.init()
screen = pygame.display.set_mode((600, 400))
pygame.display.set_caption("Random Sky - press SPACE")

sky = (135, 206, 235)

running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE:
            sky = (random.randint(0, 255),
                  random.randint(0, 255),
                  random.randint(0, 255))
            screen.fill(sky)
            pygame.display.flip()

pygame.quit()
```

Try these one by one

Sky 1 Limit the sky to dark colours only by using `randint(0, 80)`.

Sky 2 Limit it to light pastels with `randint(180, 255)`.

Sky 3 Add a yellow sun that stays at (300, 100), radius 50.

```
pygame.draw.circle(screen, (255, 220, 0), (300, 100), 50)
```

Game 6 · Hypnotic Spiral (turtle bonus)

What it does

A throwback to Lesson 1's turtle! Draws 200 line segments, each one a little longer than the last, turning 91 degrees (not 90, on purpose). Cycle through 8 colours and you get a hypnotic, multi-coloured spiral that fills the screen.

```
import turtle

# Set up the screen
screen = turtle.Screen()
screen.bgcolor("black")

# Create the turtle
pen = turtle.Turtle()
pen.speed(0)          # fastest speed
pen.width(2)

# List of colours - same idea as the Palette Test Card!
colours = ["red", "orange", "yellow", "green",
           "cyan", "blue", "purple", "magenta"]

# Draw the spiral
for i in range(200):
    pen.pencolor(colours[i % len(colours)])
    pen.forward(i * 2)
    pen.right(91)

# Keep the window open
turtle.done()
```

Notice the pattern: a list of colours, then `colours[i % len(colours)]` picks the next one each loop, wrapping around when it runs out. Same trick as the Palette Test Card and Rainbow Stripes - just with turtle instead of pygame.

Try these one by one

Spiral 1 Make it less wild - change `right(91)` to `right(90)`. The spiral becomes a clean square pattern.

Spiral 2 Try `right(120)` for a triangle-based spiral, or `right(144)` for a star.

Spiral 3 Swap the 8 colours for your own palette - use just 3 and see what happens.

```
colours = ["hotpink", "cyan", "yellow"]
```

Spiral 4 Make the spiral grow slower: change `forward(i * 2)` to `forward(i)`.

Spiral 5 Set the background to white and watch the colours pop differently.

```
screen.bgcolor("white")
```

Where these colours take us next

Every game from here uses your colour palette. In Lesson 4 we'll build the Player - a coloured rectangle you control with the arrow keys. Your palette from this lesson is what the player will be painted with.

Challenge: pick any game above, change every colour to your own palette, run it, and screenshot the result. That's your first piece of pygame art!