

# Creative Coding

Drawing order (Layering) and Basic Shapes

COD 207 - Week 03 Class →



# Table of Contents

- 1. Creative Coding
- 2. Table of Contents
- 3. Wrap-up (Summary)
- 4. Next
- 5. Structure
- 6. Coordinate System
- 7. Color (fill function)
- 8. Rendering
- 9. Rendering, Built-in & Inline Functions
- 10. Drawing Order
- 11. BREAK
- 12. *"First, solve the puzzle.*
- 13. Case Study (Piet Mondrian)
- 14. Computational Thinking Framework
- 15. Computational Thinking Framework
- 16. CT Framework Implementation
- 17. Assignments

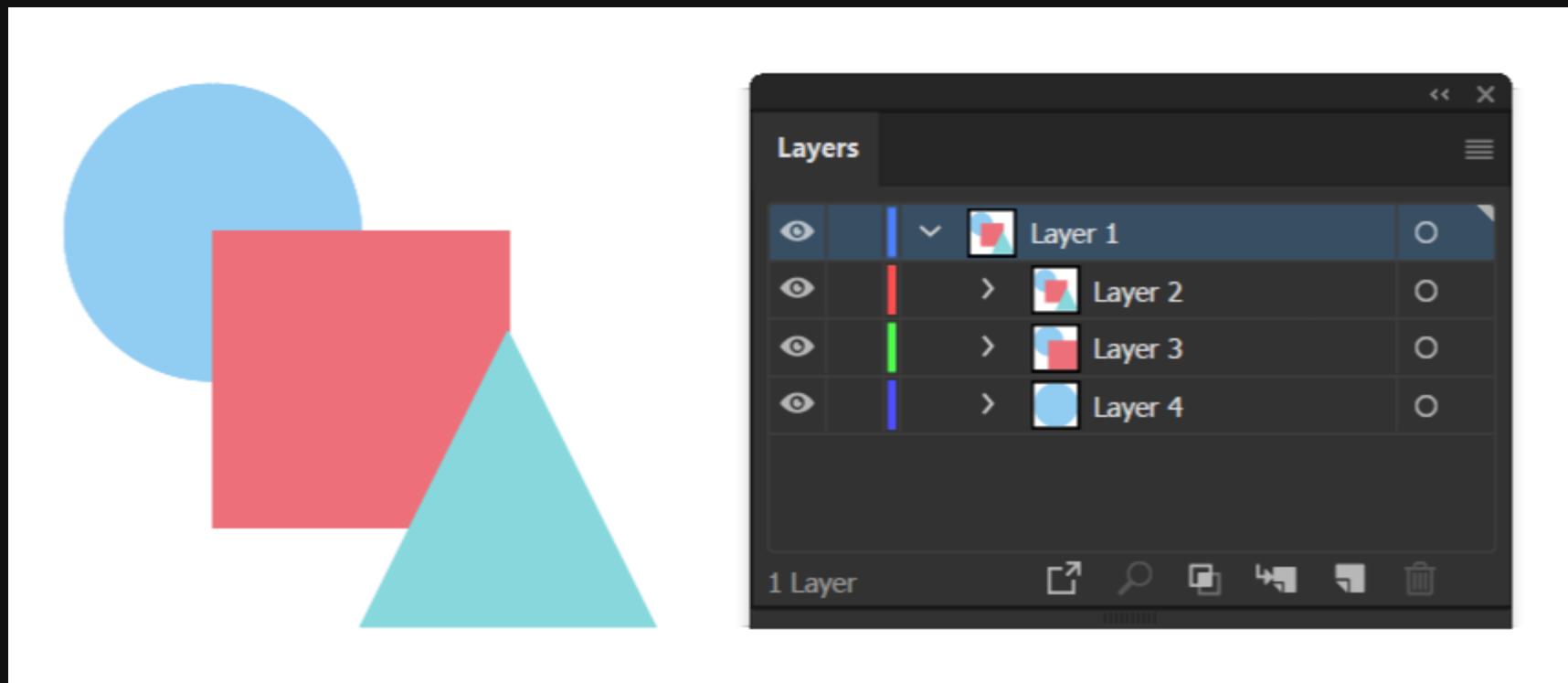
# Wrap-up (Summary)

Things we learn about P5JS programming language.

1. Structure
2. Cartesian Coordinate System (How canvas positioning works)
3. Drawing simple shapes (circle, fill)
4. Rendering

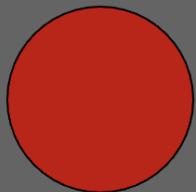
# Next

Drawing Order: Set the layer order of the shapes.



# Structure

## Inline functions



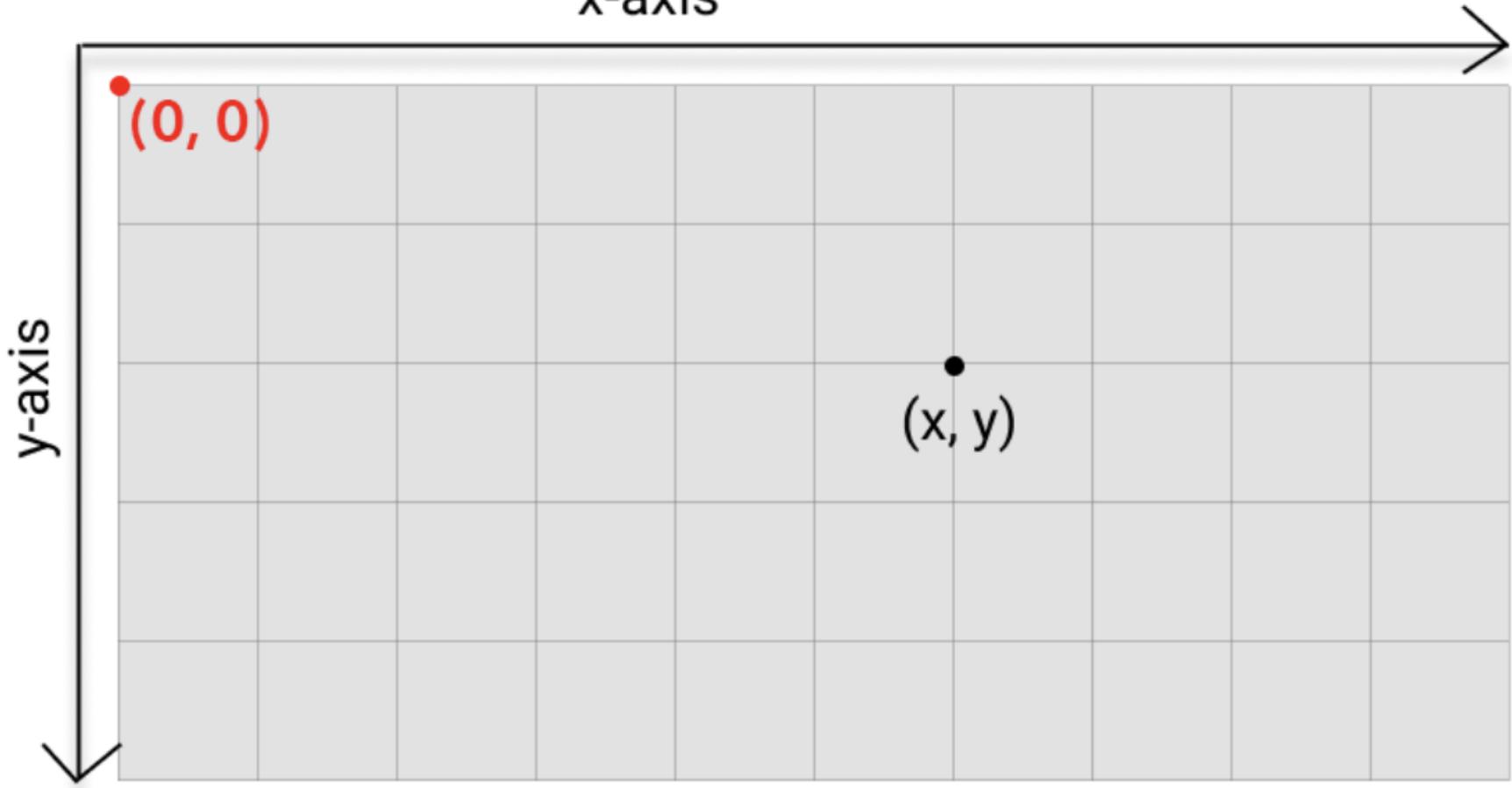
```
1  function setup() {  
2      createCanvas(400, 600);  
3  }  
4  
5  function draw() {  
6      background(100, 100, 100);  
7  
8      fill(200, 0, 0);  
9      circle(300, 300, 100);  
10     }  
11 }
```

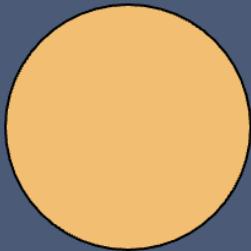
# Coordinate System

Top left corner of the the canvas is the starting point of x and y coordinates.

The p5.js canvas uses a coordinate system to describe space. The origin,  $(0, 0)$ , of the canvas is the top-left corner of the canvas.

x-axis





# Color (fill function)

A color value can be represented in various ways with p5.js. It can be given as:

- Gray value as one numeric value between 0 and 255.
- RGB (**Red**, **Green**, **Blue**) value as three numeric values between 0 and 255.

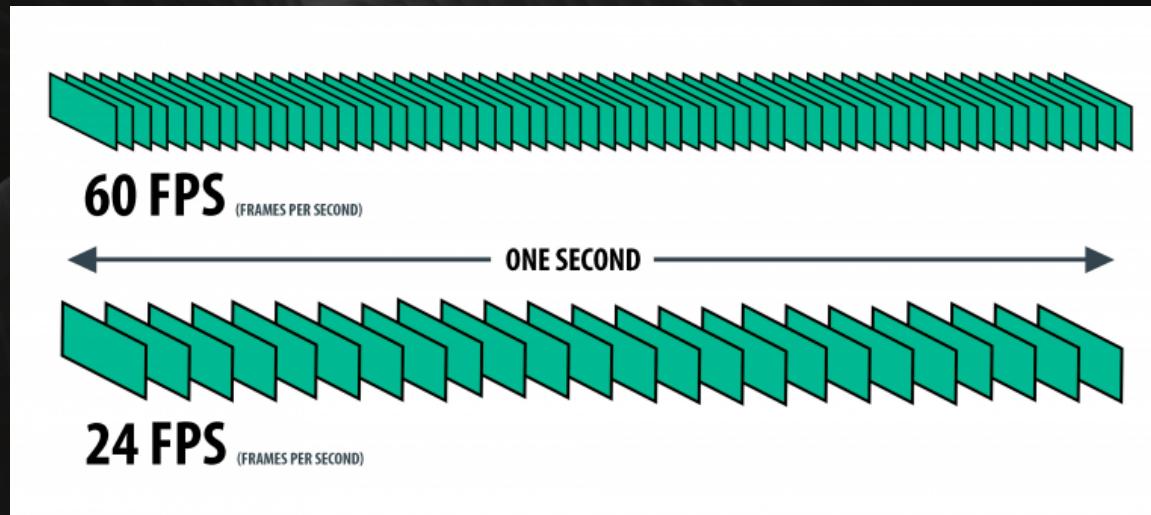
```
1  function setup() {  
2      createCanvas(400,600);  
3  }  
4  
5  function draw() {  
6      // (Red, Green, Blue)  
7      background(71,    92,     122);  
8  
9      // Uncomment the following line to change  
10     //fill(200, 10, 20);  
11     circle(252,187,100);  
12 }
```

# Rendering

Rendering in P5JS involves generating images from input data such as vector graphics such as circle, rectangle, triangle, etc...

The resulting image, also called a **render** or frame, is displayed on the user's screen multiple times per second, typically ranging from 24 to 60 frames.

The graphics is rendered 60 times per second on your canvas by default in P5JS. Click on image to get more info.



# Rendering, Built-in & Inline Functions

```
1  function setup() {
2      // Creates the application window params: width=600, height=600
3      createCanvas(600, 600);
4  }
5
6  function draw() {
7      // Set the background color of the window params: Red: 100, Green: 20, Blue: 20
8      // R,G,B values must be between 0 - 255
9      background(100,20,20);
10 }
```

# Drawing Order

Swap the drawing order of the red circle with the green one.

The order in which shape functions are called is important, as the shape function called last will be rendered on top of previously drawn shapes.

```
1  function setup() {
2      createCanvas(600, 600);
3  }
4
5  function draw() {
6      background(10, 30, 15);
7
8      // Uncomment the following line to change color
9      fill(200, 10, 20);
10     circle(300, 300, 150);
11
12     // Second circle
13     fill(0, 200, 20);
14     circle(300, 300, 150);
15 }
```



# BREAK

10 mins.

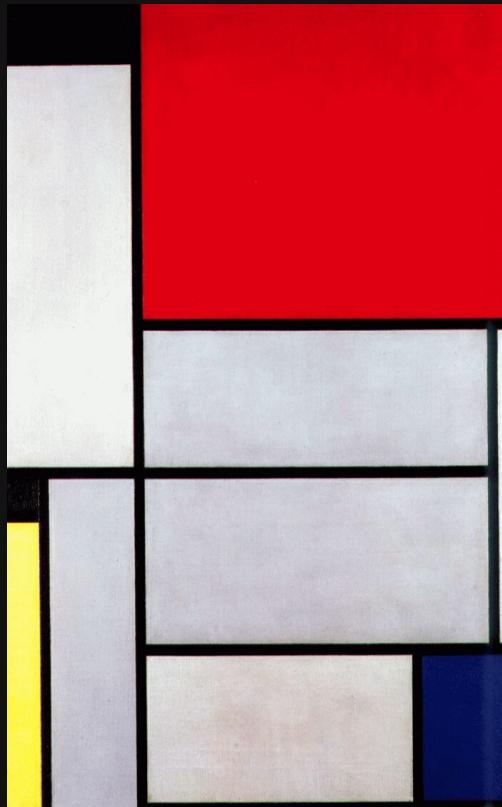
*"First, solve the puzzle.*

*Then, write the code."*

John Johnson

# Case Study (Piet Mondrian)

Era of expressionism known as 'De Stijl' (1917-1931)

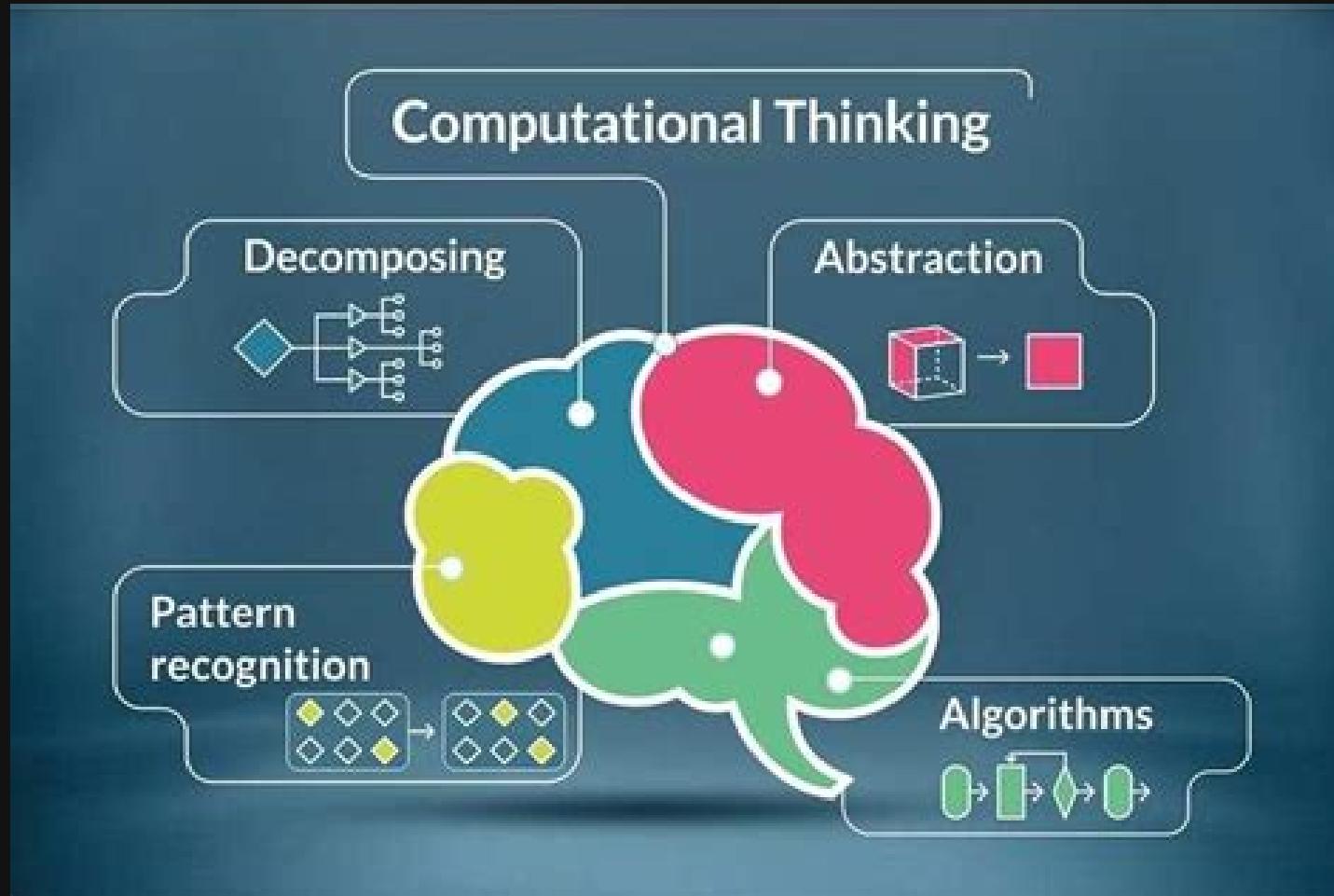


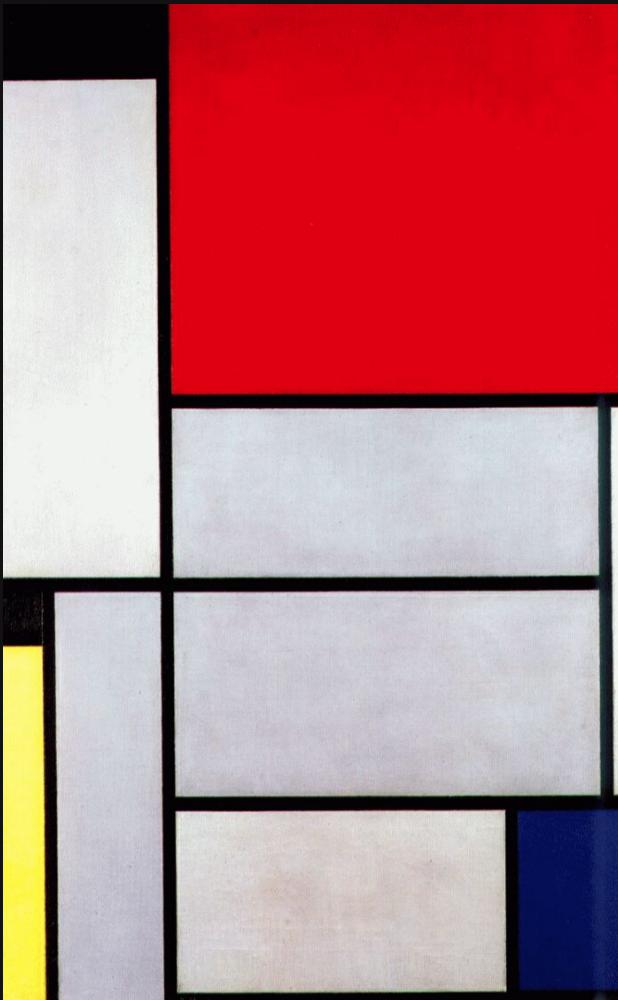
# Computational Thinking Framework

Computational thinking is a problem-solving process that includes; Decomposing, Abstraction, Pattern Recognition, Algorithms

- Formulating Problems & Analyze
- Represent data through Abstraction
- Algorithmic Thinking: Automate solutions
- Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources
- Generalizing and transferring this problem-solving process to a wide variety of problems

# Computational Thinking Framework





# CT Framework Implementation

1. Analyze:
  - Determine the color palette
  - Determine the primitive shapes (How many layers of shapes are there in the canvas?)
  - Determine the size and x-y coordinate of each shape.
2. Recall what you leaned from the last week. Which inline-function you need to use?
3. Note down the steps that is required to draw the image.

# Assignments

1. Review the Algorithmic Art works (Frieder Nake, Vera Molnar, Paul Klee, Sol LeWitt, Piet Mondrian, Russian Constructivism). Bring the artwork image to the class.
2. Read the attached document → [week4-Reading\\_Styling\\_Shape\\_Properties\\_2.pdf](#)
3. Finalize Piet Mondrian's artwork that you started coding in class.
4. Create 10 different variations of the same artwork. Change colors, add more shapes, adjust sizes of the shapes, etc...
5. You are required to submit 10 different links to the Learning Management System (LMS).
6. Additionally, please upload the source code as a ZIP file. You only need to submit one of them; there is no need to upload all of the source codes for each variation.