

# Creative Coding

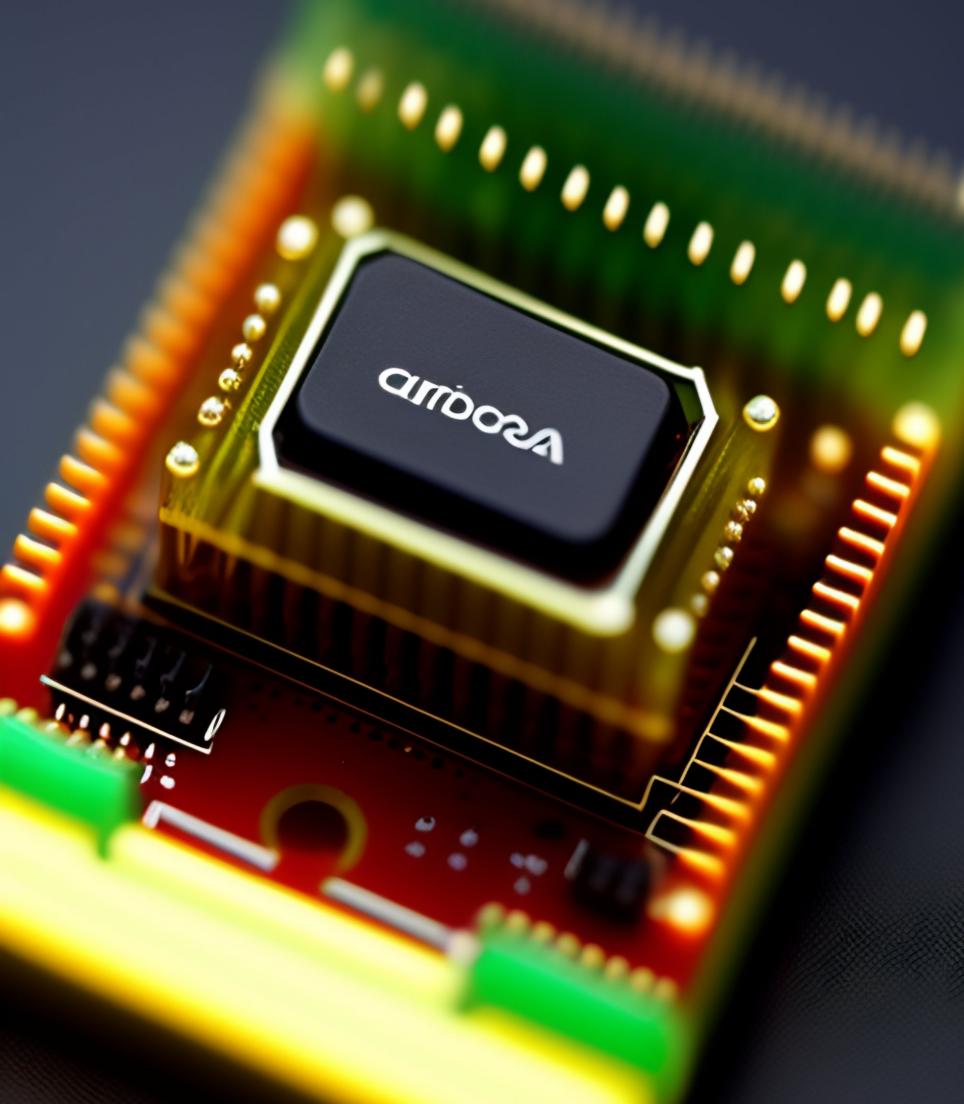
Interactivity via Computer Mouse and Keyboard

COD 207 - Week 07 Class →



# Table of Contents

- 1. Creative Coding
- 2. Table of Contents
- 3. Wrap-up (Summary)
- 4. Generative Art: Tiling to Tesselation
- 5. Generative Art: Semi-Regular Tesselation
- 6. Generative Art: Demi-Regular Tesselation
- 7. Tiling-Tesselation Repository
- 8. Built-in Functions: Keyboard
- 9. Built-in Functions: Mouse
- 10. BREAK
- 11. Nested For Loops: Tiling
- 12. Download The CheatSheet
- 13. Assignments



# Wrap-up (Summary)

Things we learn about P5JS programming language.

- Variables
- Conditionals (If/else)
- Loops (for)

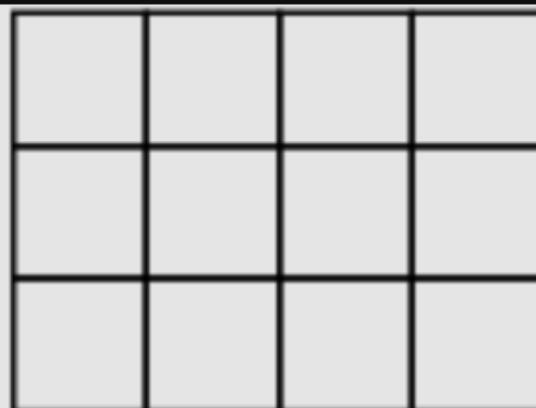
# Generative Art: Tiling to Tesselation

Tiling is repeating a pattern in horizontal, vertical or both directions repeatedly covering any kind of surface.

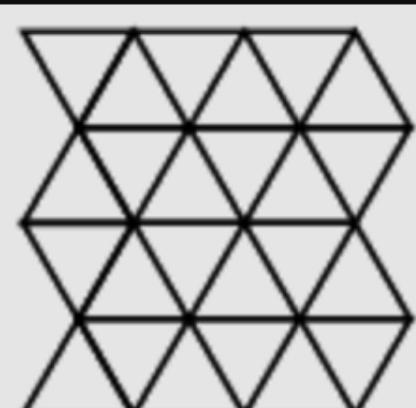
Regular Tessellation



{6, 3}



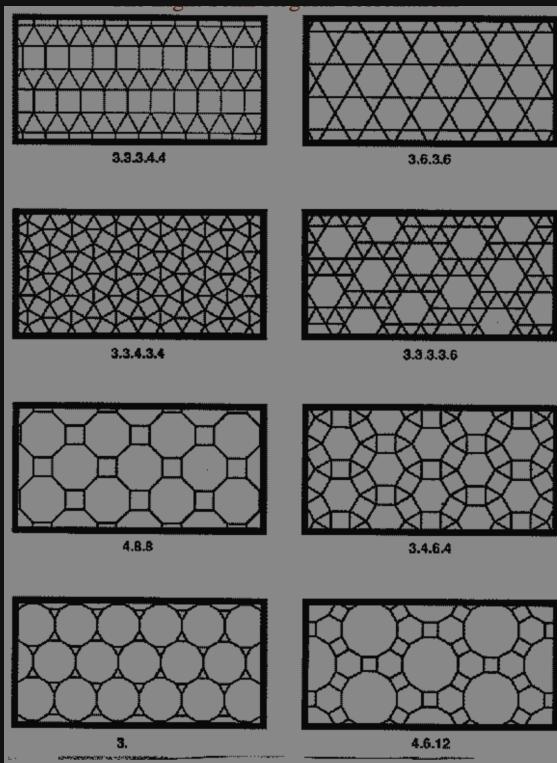
{4, 4}



{3, 6}

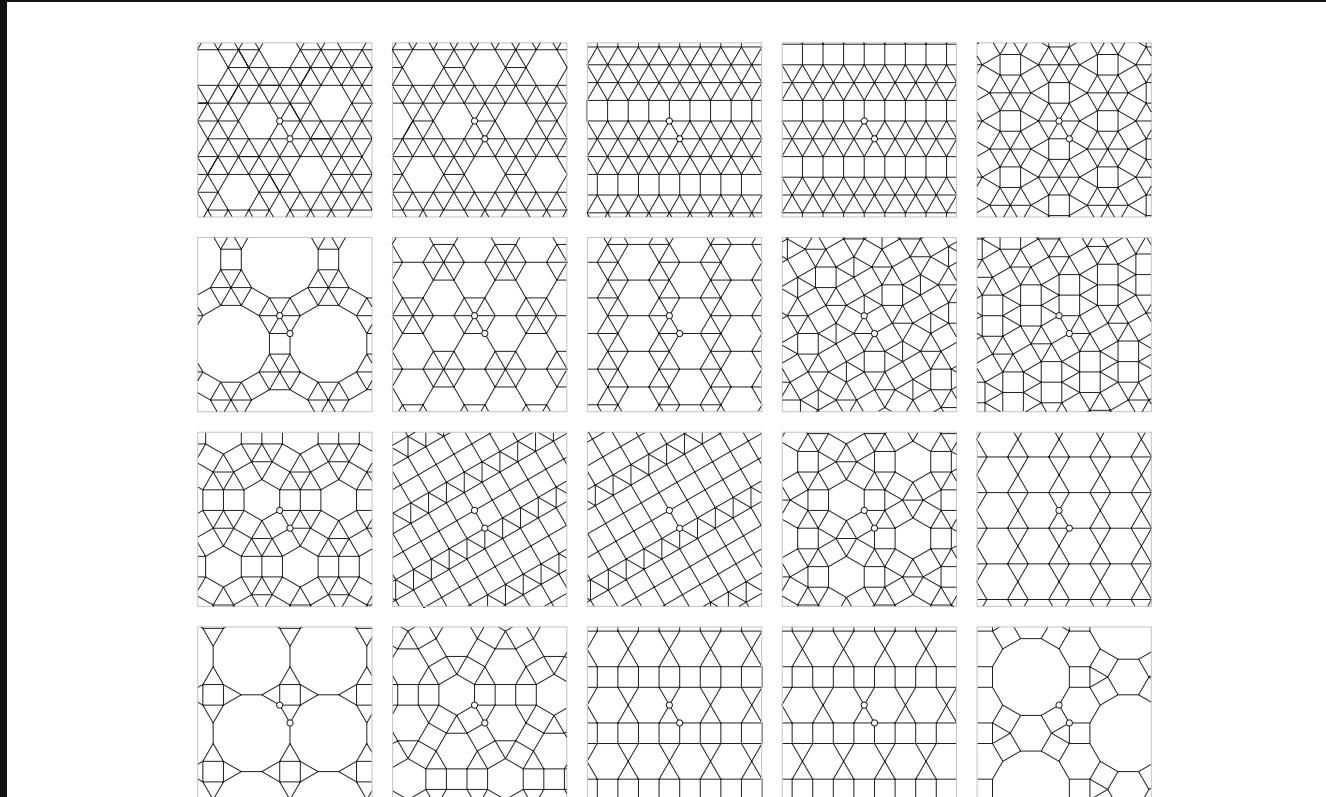
# Generative Art: Semi-Regular Tesselation

## Semi-Regular Tesselation

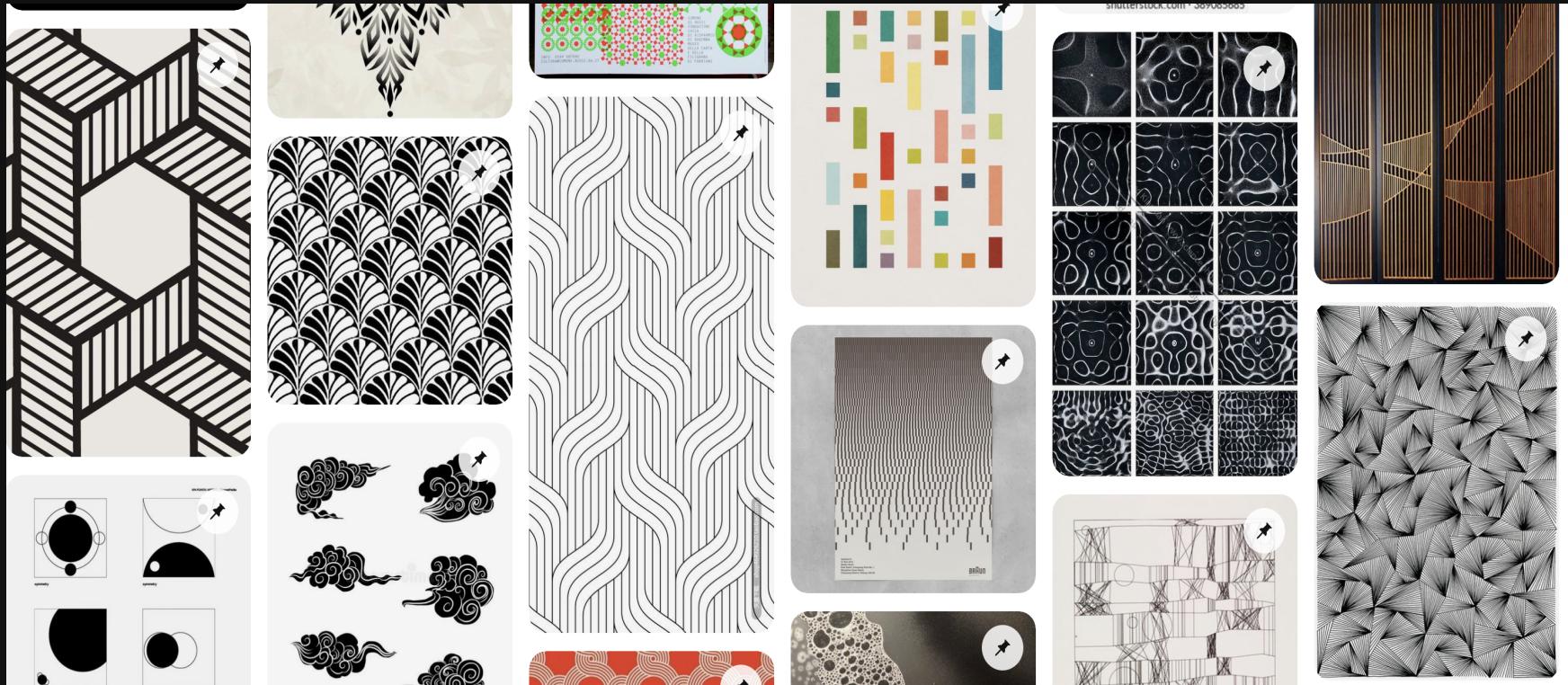


# Generative Art: Demi-Regular Tessellation

## Demi-Regular Tessellation



# Tiling-Tesselation Repository



 Goto Repository

# Built-in Functions: Keyboard

```
1  function setup() {  
2      // Creates the application window params: width=600, height=600  
3      createCanvas(600, 600);  
4      noLoop();  
5  }  
6  
7  function draw() {  
8      // Set the background color of the window params: Red: 100, Green: 20, Blue: 20  
9      // R,G,B values must be between 0 - 255  
10     background(100,20,20);  
11  }  
12  
13 // key is built-in variable. You don't need to declare on top of your code.  
14 function keyPressed() {  
15     if(key == 'r') {  
16         // set the background to a random color  
17     }  
18 }
```

# Built-in Functions: Mouse

```
1  function setup() {  
2      // Creates the application window params: width=600, height=600  
3      createCanvas(600, 600);  
4      noLoop();  
5  }  
6  
7  function draw() {  
8      // Set the background color of the window params: Red: 100, Green: 20, Blue: 20  
9      // R,G,B values must be between 0 - 255  
10     background(100,20,20);  
11     circle(width/2,height/2,300);  
12  }  
13  
14  // this function will be triggered whenever the user clicks on mouse or touchpad  
15  function mousePressed() {  
16      // Increase the size of the circle whenever the user clicks on mouse or touchpad  
17  }
```



# BREAK

10 mins.

# Nested For Loops: Tiling

```
1  function setup() {
2      // Creates the application window params: width=600, height=600
3      createCanvas(600, 600);
4      noLoop();
5  }
6
7  function draw() {
8      // Set the background color of the window params: Red: 100, Green: 20, Blue: 20
9      // R,G,B values must be between 0 - 255
10     background(100,20,20);
11
12     for(let i = 0; i < 6; i++ ) {
13         for(let j = 0; j < 6; j++ ) {
14             let x = i * 100;
15             let y = j * 100;
16             rect(x, y, 100, 100);
17         }
18     }
19 }
```

# Download The CheatSheet

Cheatsheets / Learn p5.js



## Interaction

### mouseX and mouseY

The `mouseX` and `mouseY` variables always store the current x and y coordinates of the mouse relative to the origin of the canvas. So if the mouse was currently at the x position of 150 pixels and the y position of 200 pixels, the value of the `mouseX` variable would be 150 and the value of the `mouseY` variable would be 200.

```
function draw() {  
    // The ellipse's x and y positions  
    // follow the mouse  
    ellipse(mouseX, mouseY, 100, 100);  
}
```

### The mouseIsPressed Variable

`mouseIsPressed` is a built-in boolean variable that is `true` when the mouse button is pressed, and `false` when it is not pressed. The `mouseIsPressed` variable is commonly used in `if` statements to perform actions based on whether the mouse button has been pressed or not.

```
// Draws ellipse if mouse is pressed  
if (mouseIsPressed) {  
    ellipse(200, 200, 180, 180);  
}
```

[Download The CheatSheet ](Learn p5.js\_ Interaction Cheatsheet \_ Codecademy.pdf)

# Assignments

1. Check the online repository. Choose a design that includes tiling.
2. Remake it using P5JS.
3. Analyze and write down your steps (Computational Thinking Framework). PTS
4. You must add comments to your sketch. PTS
5. You must use nested for loops. PTS
6. You must use randomness. PTS
7. The sketch must be interactive via keyboard or mouse. PTS
8. Submit the openprocessing link. PTS
9. Submit the sketch source code as zip file as well. PTS