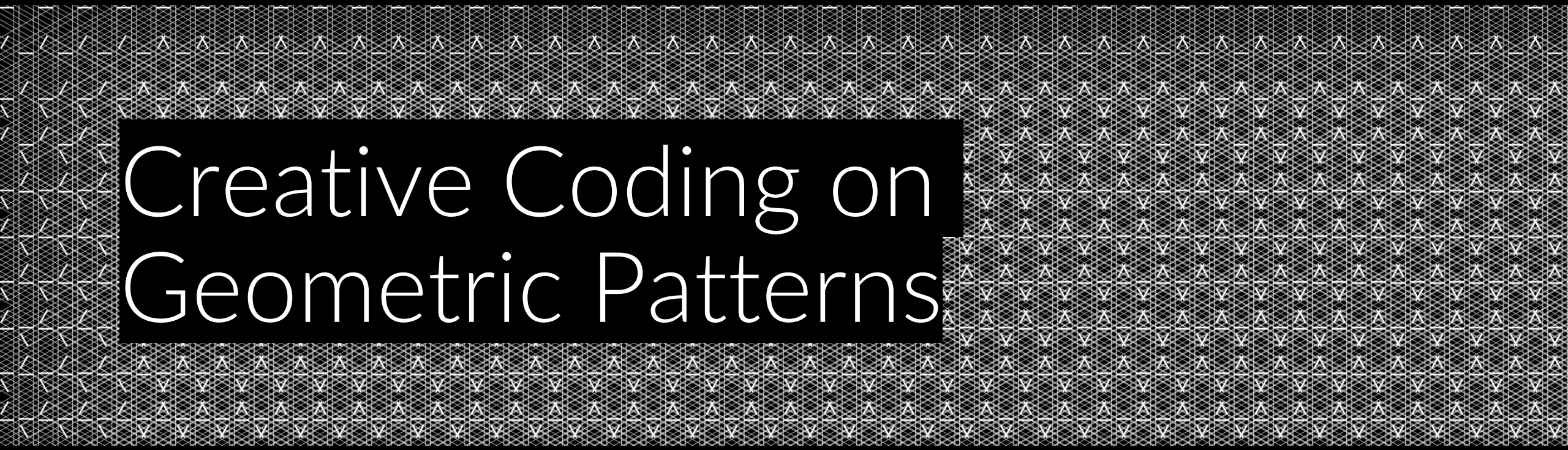


Creative Coding on Geometric Patterns

Presenter: Assoc. Prof. Dr. Selcuk ARTUT

Visual Arts and Visual Communication Design

Sabanci University, Istanbul Turkiye

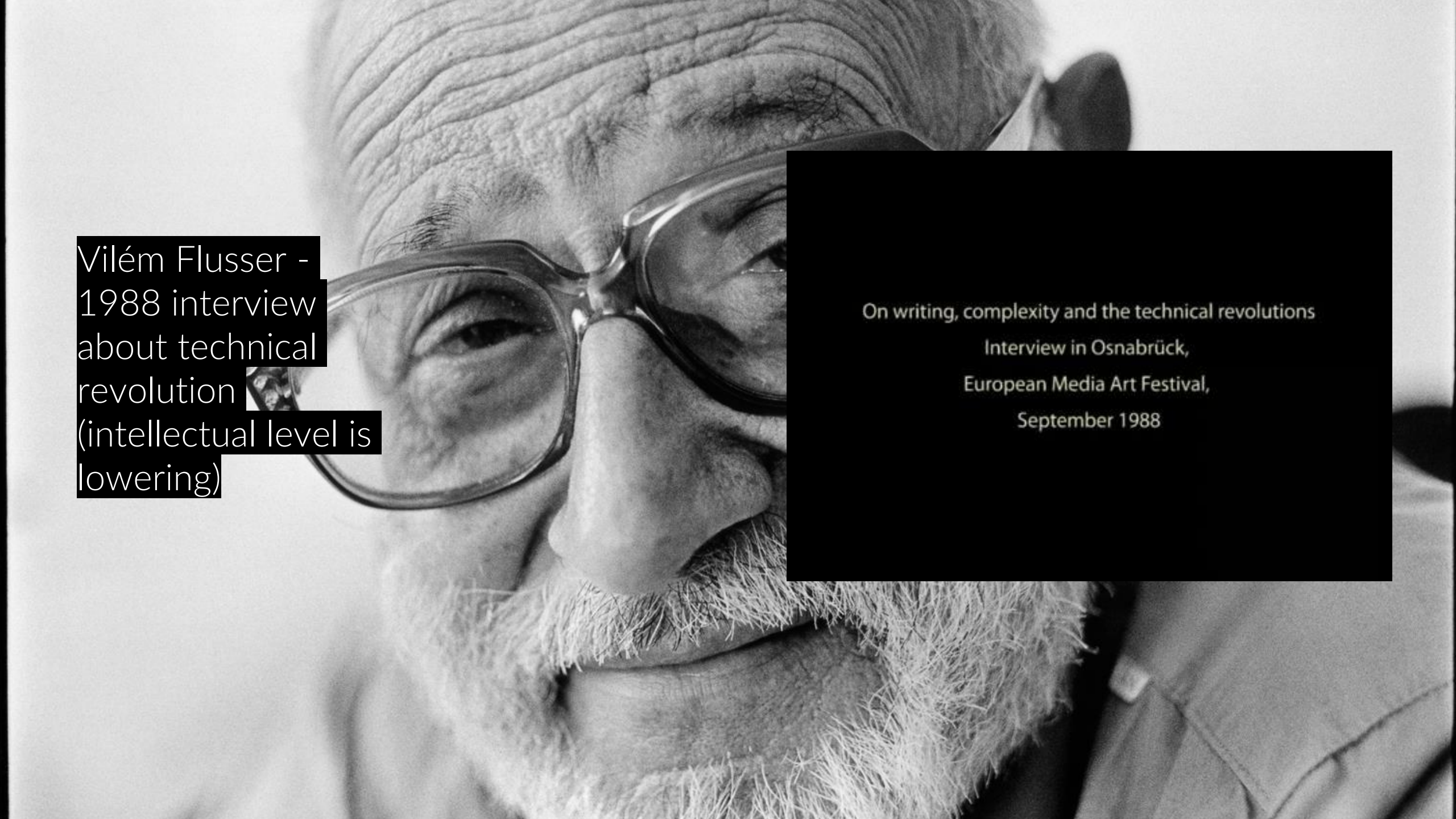
Creativity

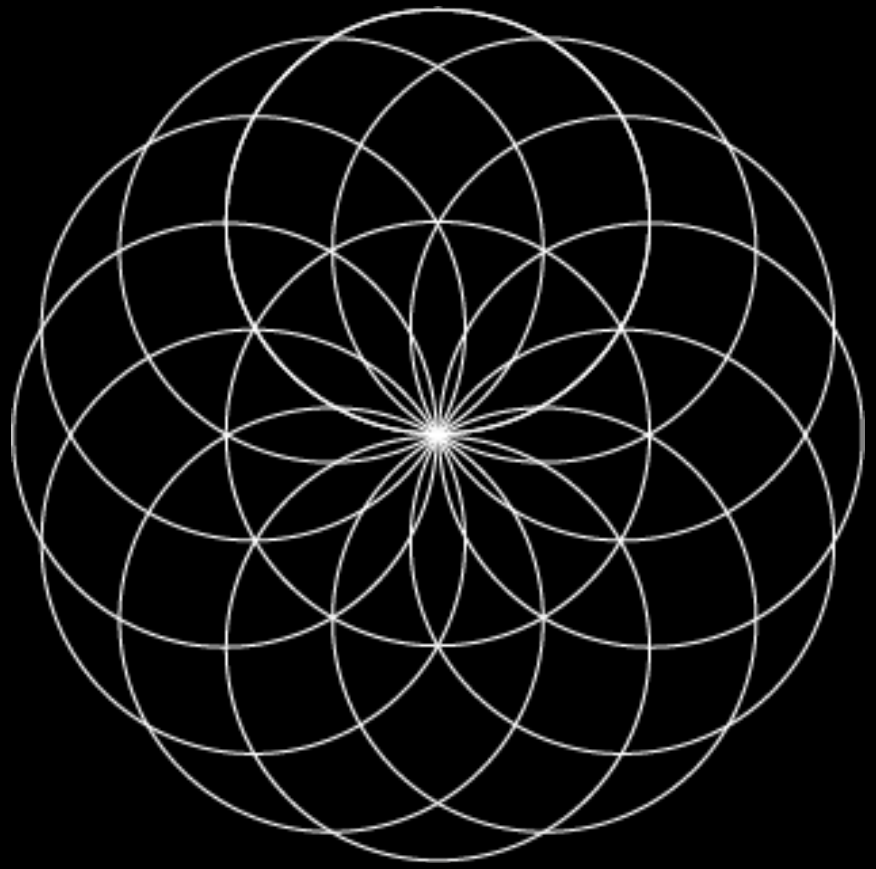
Creativity is a phenomenon whereby something new and somehow valuable is formed. The created item may be intangible (such as an idea, a scientific theory, a musical composition, or a joke) or a physical object (such as an invention, a literary work, or a painting).

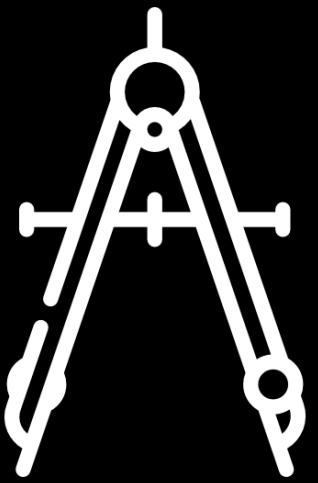
Creative coding is a type of computer programming in which the goal is to create something expressive instead of something functional.

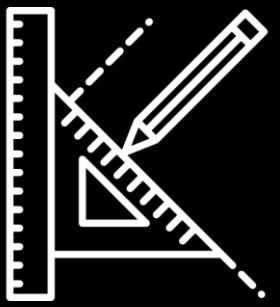
WIKIPEDIA
The Free Encyclopedia

Vilém Flusser - 1988 interview about technical revolution (intellectual level is lowering)









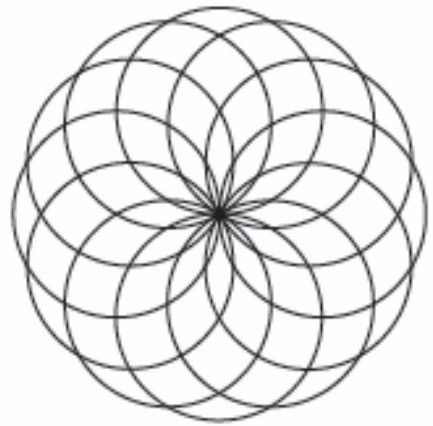
Hands on tutorial

Creating Symmetric Geometric Shapes

Creating Symmetric Geometric Shapes
Selcuk ARTUT

Division of a Circle in 6 & 12 equal sections

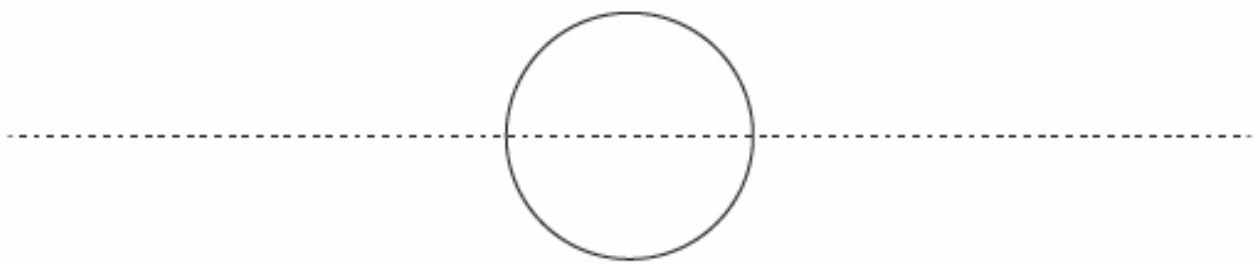
Step 1 : Draw a straight line with a pencil



Creating Symmetric Geometric Shapes
Selcuk ARTUT

Division of a Circle in 6 & 12 equal sections

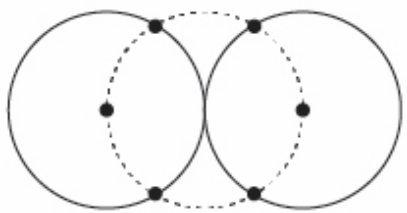
Step 2 : Use compass to make a circle, place its center on the line



Creating Symmetric Geometric Shapes
Selcuk ARTUT

Division of a Circle in 6 & 12 equal sections

Step 3 : Draw another circle to right, center its origin adjacent to previous circle

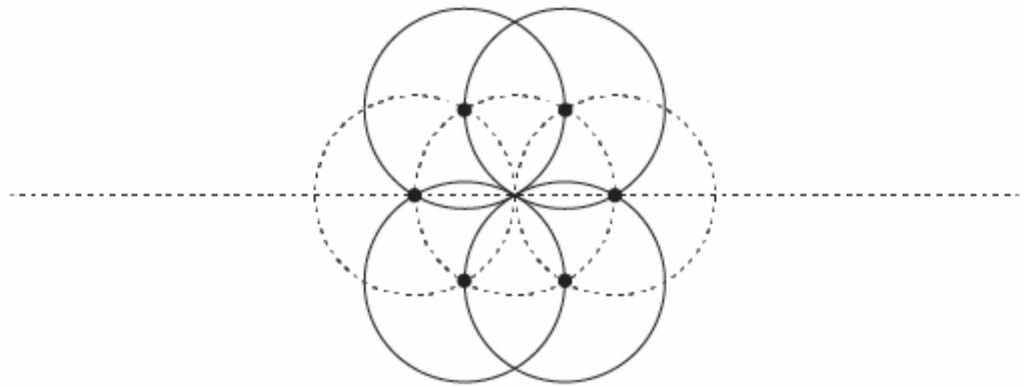


Now you have 6 sections

Creating Symmetric Geometric Shapes
Selcuk ARTUT

Division of a Circle in 6 & 12 equal sections

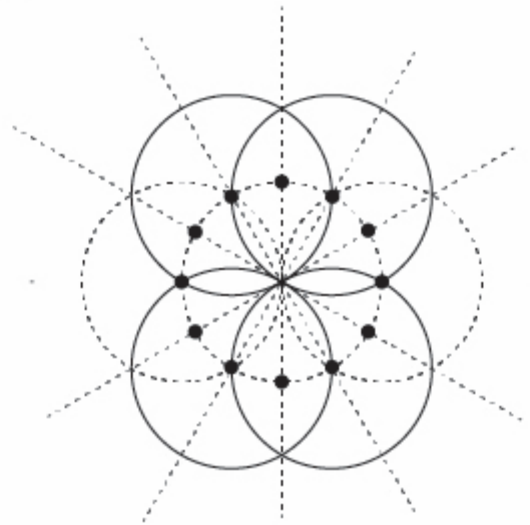
Step 4 : Draw circles on intersection points



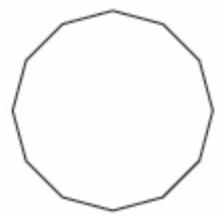
Creating Symmetric Geometric Shapes
Selcuk ARTUT

Division of a Circle in 6 & 12 equal sections

Step 5 : Draw lines across the intersection points



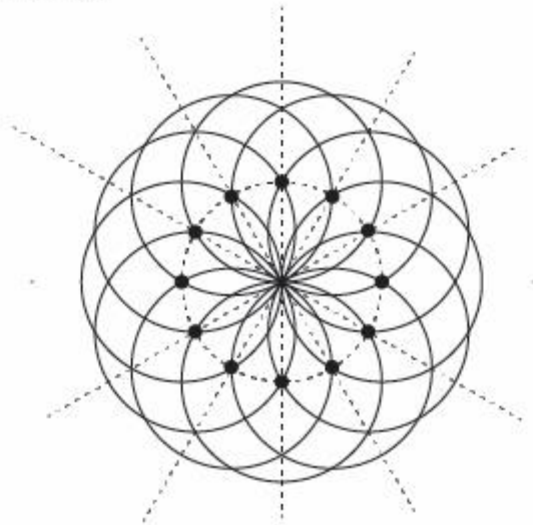
Now you have 12 sections



Creating Symmetric Geometric Shapes
Selcuk ARTUT

Division of a Circle in 6 & 12 equal sections

Step 5 : Draw circles centered at the intersection points

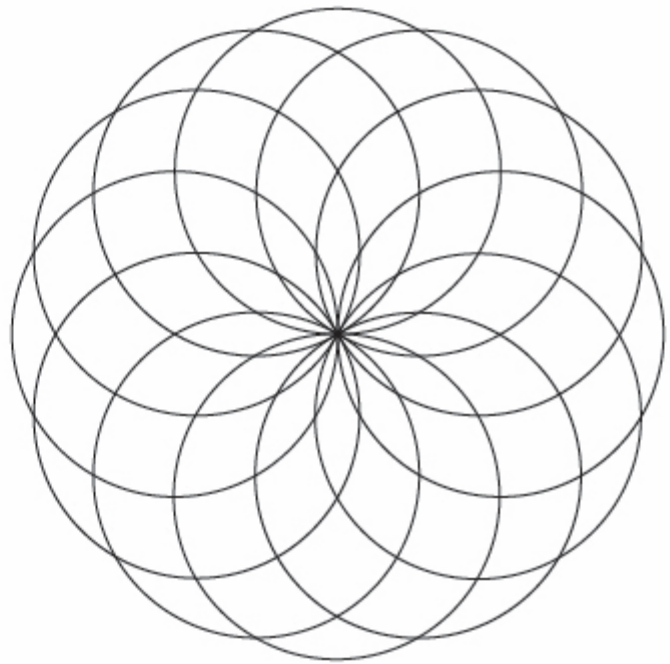


Creating Symmetric Geometric Shapes
Selcuk ARTUT

Illustrator Method

To rotate around a different reference point, select the Rotate tool. Then Alt-click (Windows) or Option-click (Mac OS) where you want the reference point to be in the document window.

To rotate around the center point, choose Object > Transform > Rotate, or double-click the Rotate tool.



Most Common Creative Coding Platforms

Processing

Openframeworks

Cinder

p5.js

Touchdesigner

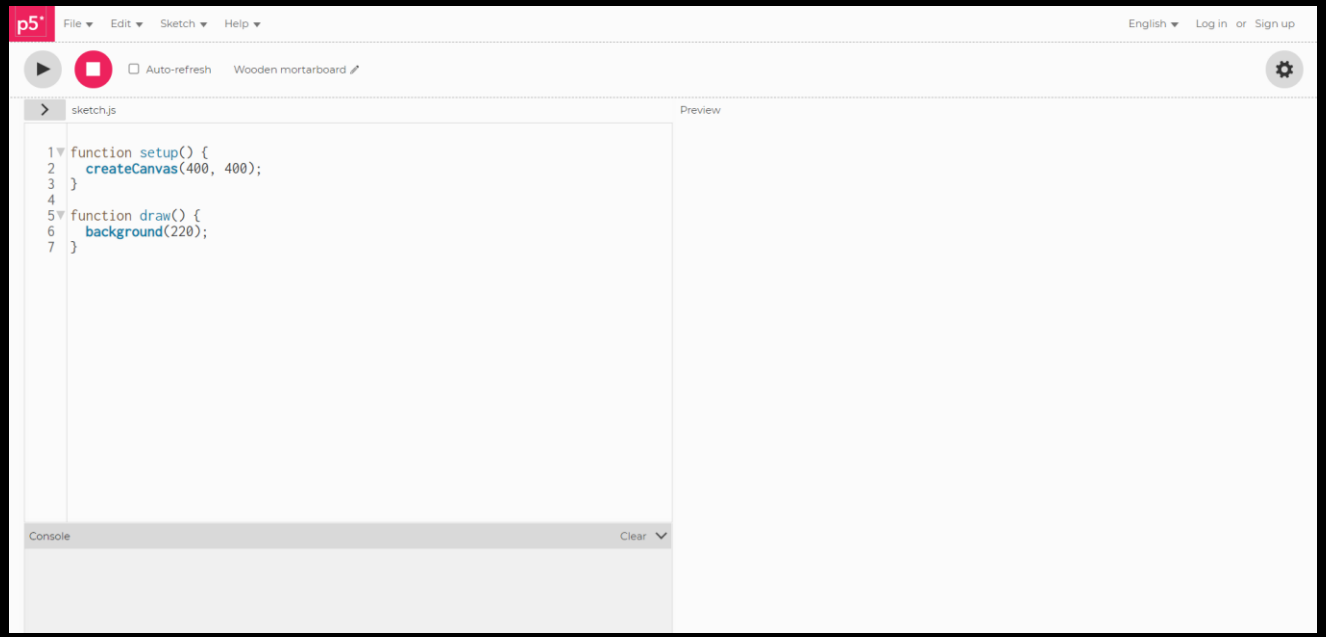
MaxMSP/Jitter

Vvvv

p5.js is a JavaScript library for creative coding, with a focus on making coding accessible and inclusive for artists, designers, educators, beginners, and anyone else! p5.js is free and open-source because we believe software, and the tools to learn it, should be accessible to everyone.

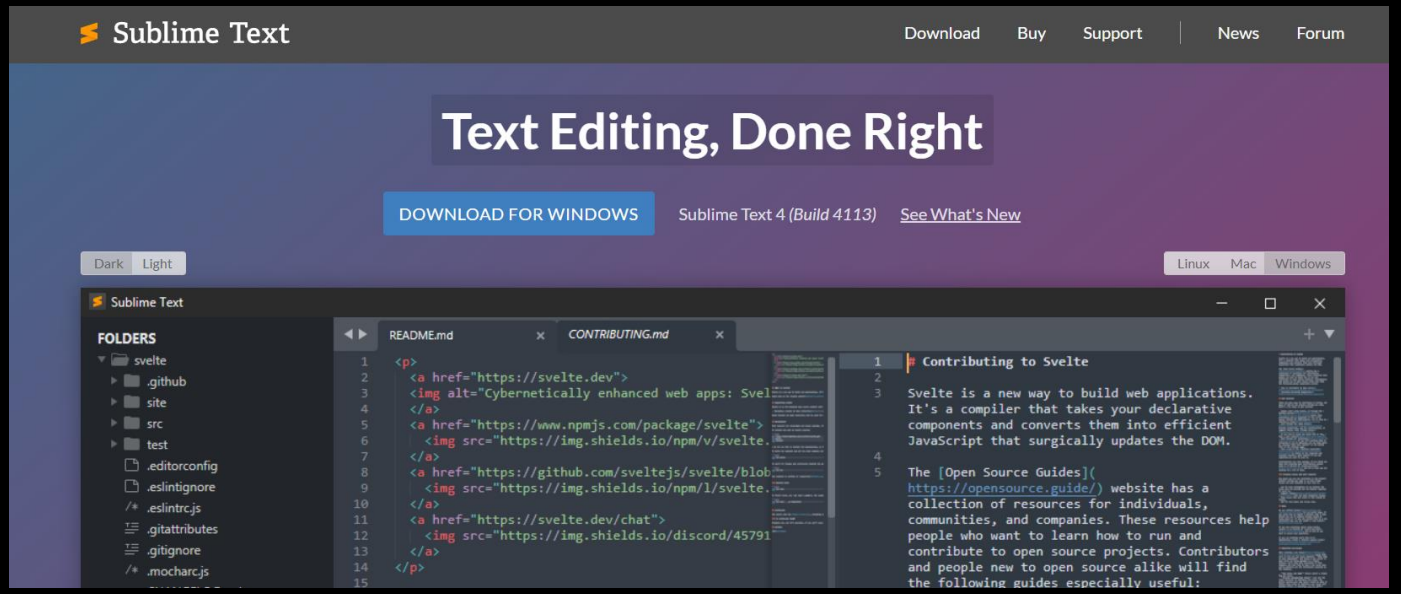
Using p5.js

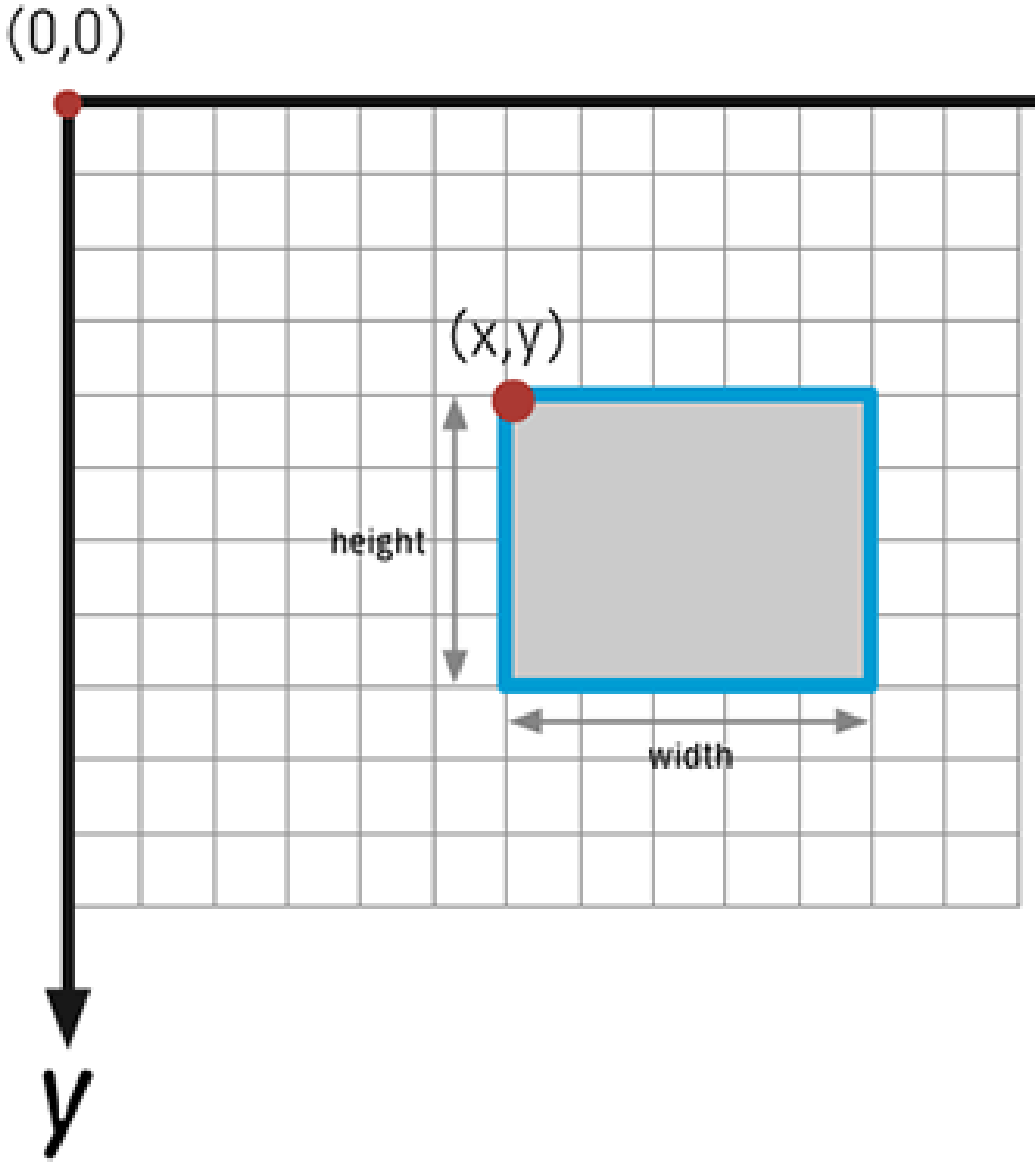
https://p5js.org/



Using p5.js

https://www.sublimetext.com/





rect(x,y,width,height)

The coordinate origin (0,0) is at the top left of the display window. Increasing *x* moves right, increasing *y* moves down.

Analyzing p5.js Coding Environment

```
function setup() {
        createCanvas(400, 400);
}

function draw() {
        background(220);
}
```

Syntax

```
createCanvas(w, h, [renderer])
```

Parameters

w	Number: width of the canvas
h	Number: height of the canvas
renderer	Constant: either P2D or WEBGL (Optional)

Using Comments

Comments are to be used by humans only. So computer will ignore those lines, and compile the uncommented lines.

Comment Types

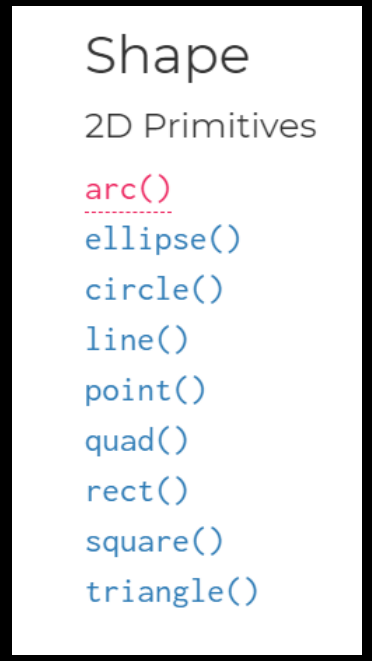
Block Comment : /* ..... */ (Using forward slash asterix .... asterix forward slash)

Line Comment : // (Using two forward slash)

Alternative usage : You may want to ignore a block or a line of code to check for debugging.

Drawing Basic Shapes

Some simple shapes;
point : https://p5js.org/reference/#/p5/point
circle: https://p5js.org/reference/#/p5/circle
ellipse : https://p5js.org/reference/#/p5/ellipse
rect : https://p5js.org/reference/#/p5/rect
square: https://p5js.org/reference/#/p5/square
line : https://p5js.org/reference/#/p5/line
arc : https://p5js.org/reference/#/p5/arc
triangle : https://p5js.org/reference/#/p5/triangle



Important note: in 2D mode (i.e. when p5.Renderer is not set) the origin (0,0) is positioned at the top left of the screen. In 3D mode (i.e. when p5.Renderer is set to WEBGL), the origin is positioned at the center of the canvas.

Drawing Basic Shapes

```
createCanvas(100, 100, WEBGL);
background(240, 240, 240);
fill(237, 34, 93);
noStroke();
beginShape();
vertex(-10, 10);
vertex(0, 35);
vertex(10, 10);
vertex(35, 0);
vertex(10, -8);
vertex(0, -35);
vertex(-10, -8);
vertex(-35, 0);
endShape();
```

Colors

color() : Creates colors for storing in variables of the color datatype. The parameters are interpreted as RGB or HSB values depending on the current colorMode(). The default mode is RGB values from 0 to 255 and, therefore, color(255, 204, 0) will return a bright yellow color.

Syntax

color(gray)

color(gray, alpha)

color(v1, v2, v3)

color(v1, v2, v3, alpha)

https://p5js.org/reference/#/p5/color

```
fill(255, 204, 0);
noStroke();
rect(30, 20, 55,
```

Variables

In order to create dynamic structures, we are required to have data carriers that can change its value at a particular moment. So variables are **great** tools for **Creative Coding.**

Variable Examples

```
let a = 50;

let t = "hello";

let boo = true;

let col = color(255, 204, 0);
```

Loops Example:

```
for (let i = 0; i < 9; i++) {
  console.log(i);
}
```

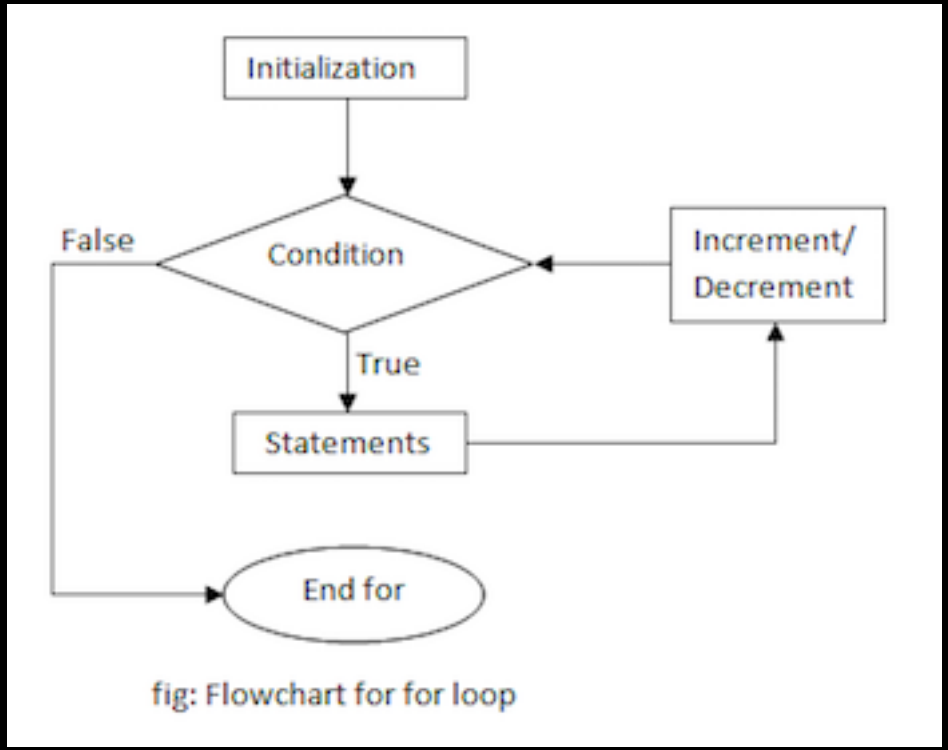
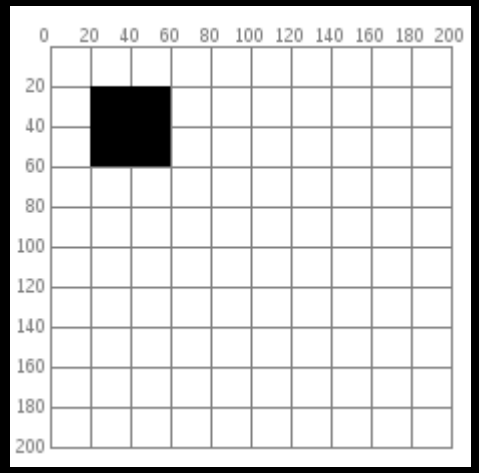


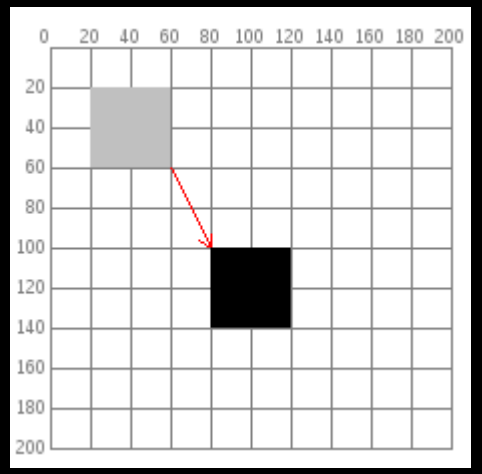
fig: Flowchart for for loop

Introduction to The Transformation Matrix

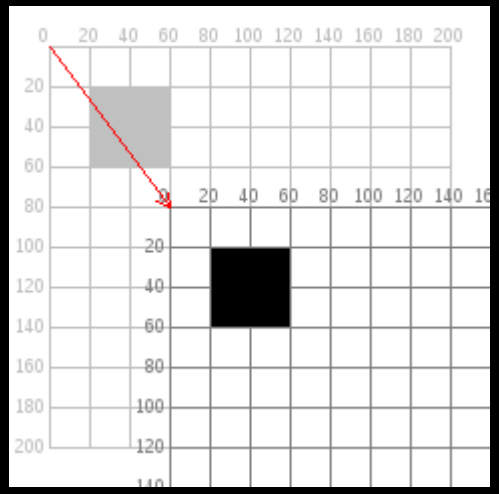
As you know, your p5.js window works like a piece of graph paper. When you want to draw something, you specify its coordinates on the graph. Here is a simple rectangle drawn with the code rect(20, 20, 40, 40). The coordinate system (a fancy word for “graph paper”) is shown in gray.



If you want to move the rectangle 60 units right and 80 units down, you can just change the coordinates by adding to the x and y starting point: rect(20 + 60, 20 + 80, 40, 40) and the rectangle will appear in a different place. (We put the arrow in there for dramatic effect.)



But there is a more interesting way to do it: move the graph paper instead. If you move the graph paper 60 units right and 80 units down, you will get exactly the same visual result. Moving the coordinate system is called translation.

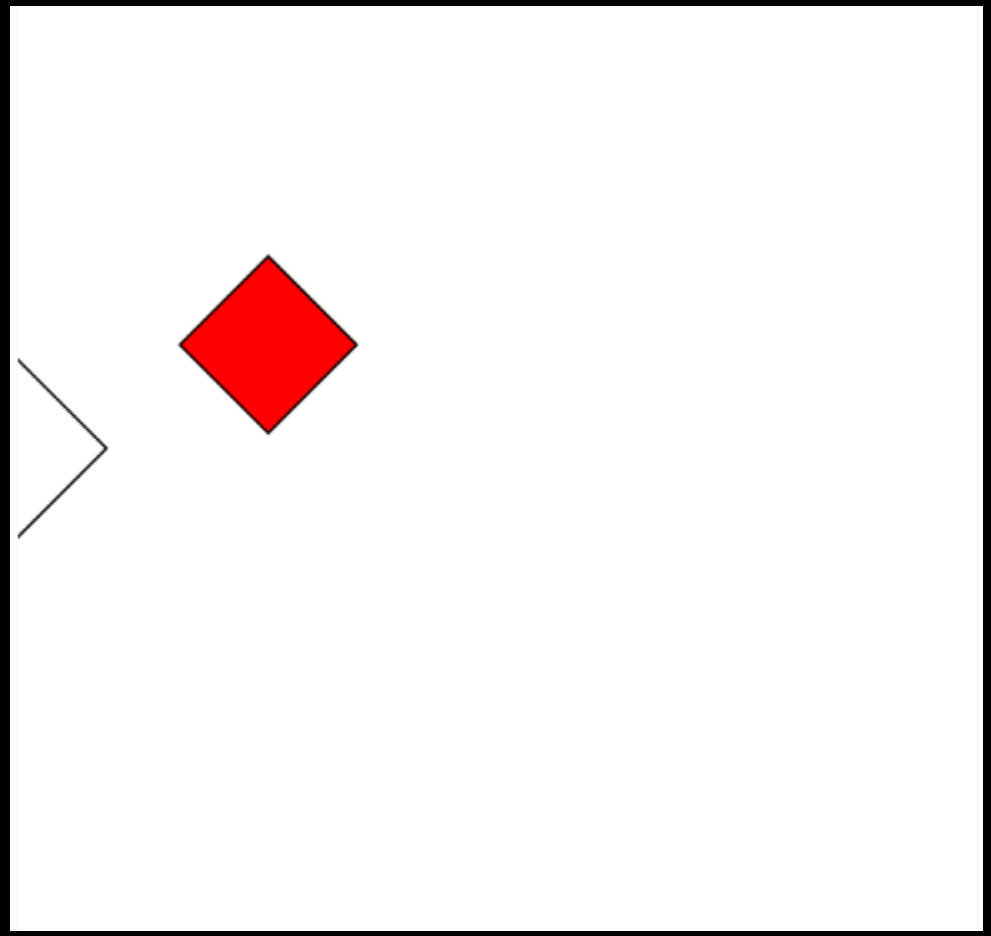


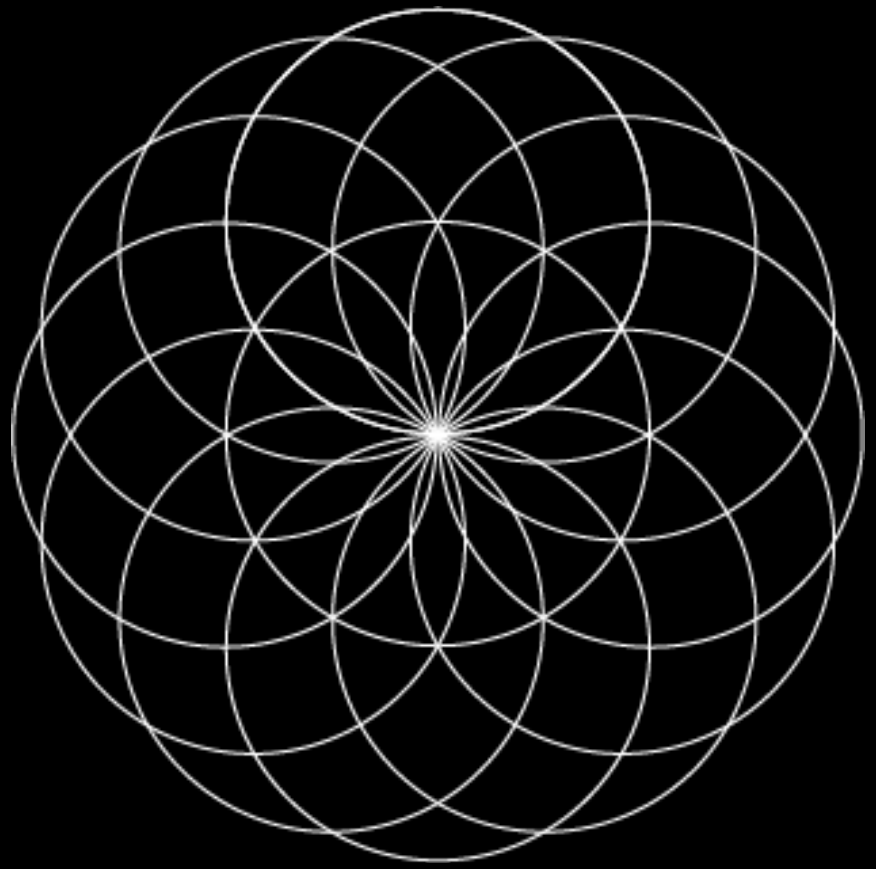
The Transformation Matrix

Every time you do a rotation, translation, or scaling, the information required to do the transformation is accumulated into a table of numbers. This table, or matrix has only a few rows and columns, yet, through the miracle of mathematics, it contains all the information needed to do any series of transformations. And that's why the push() and pop() have that word in their name.

The Transformation Matrix

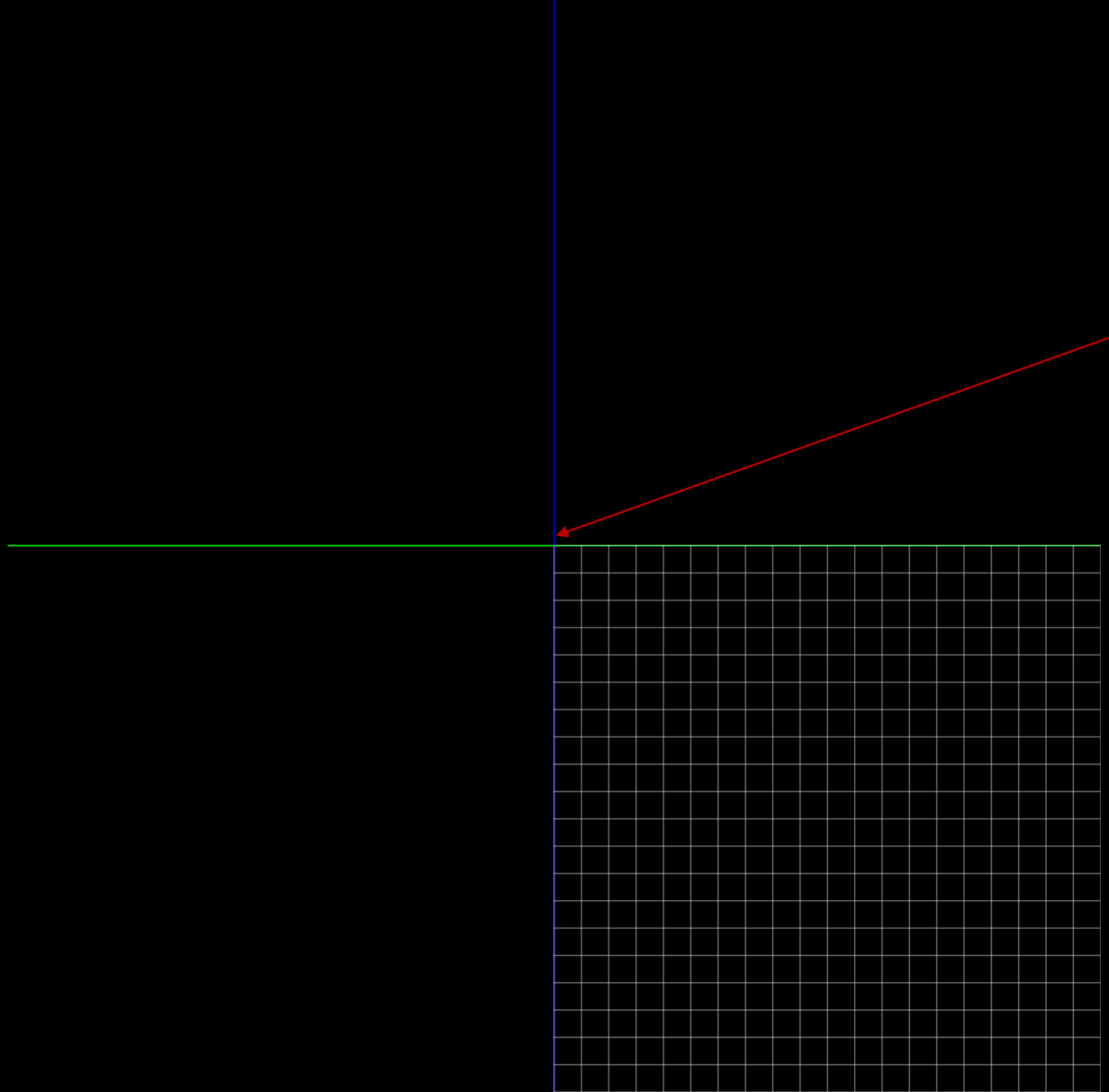
```
function setup() {
	createCanvas(400, 400);
	angleMode(DEGREES);
}
function draw() {
	background(255);
	noFill();
	push();
	rotate(45);
	translate(100, 100);
	rect(0, 0, 50, 50);
	pop();
	fill(color(255, 0, 0));
	push();
	translate(100, 100);
	rotate(45);
	rect(0, 0, 50, 50);
	pop();
}
```





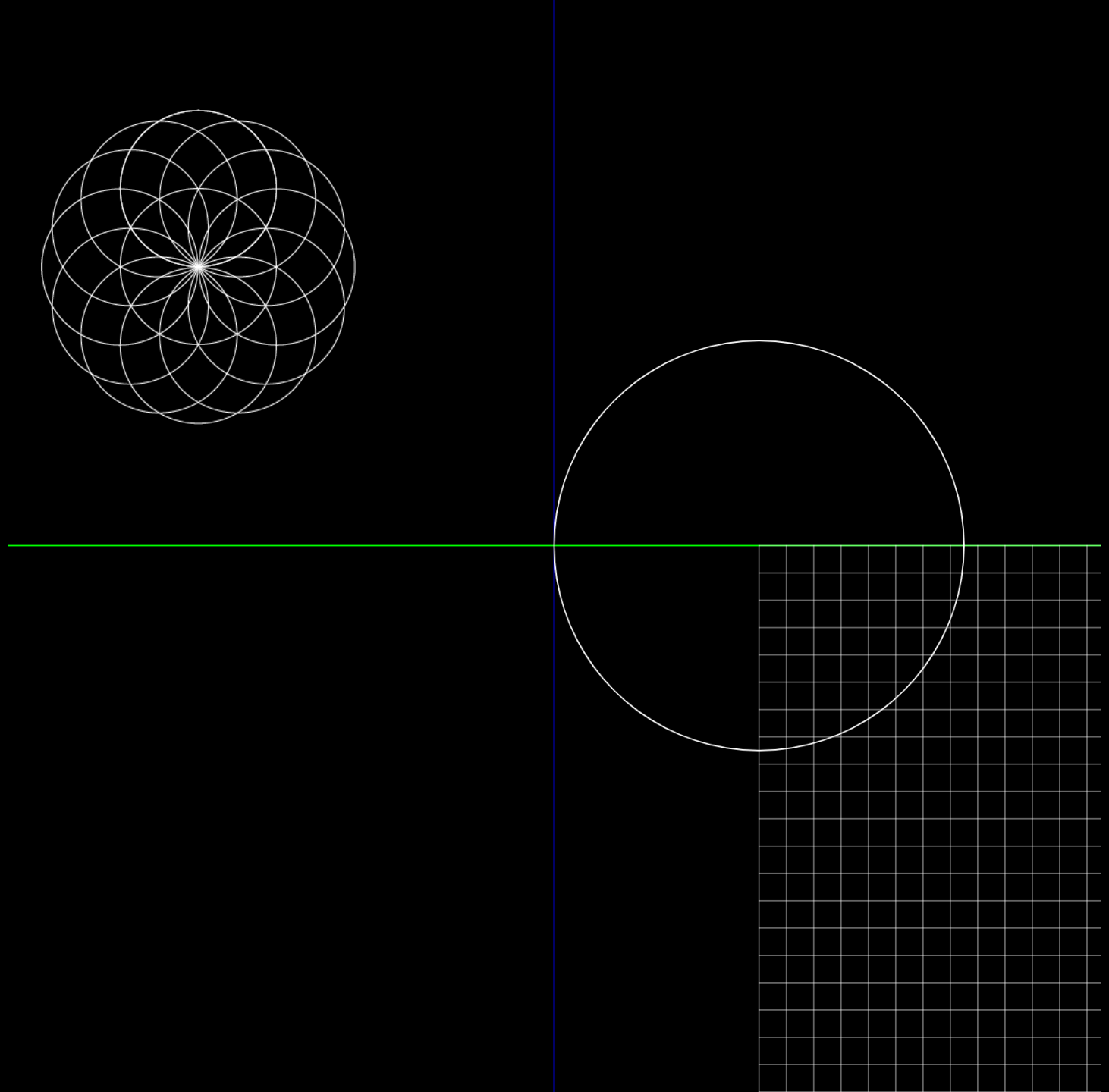
Hands on tutorial

Creating Symmetric Geometric Shapes



In order to center the image, let us translate the canvas to the center with a transformation

```
push();
translate(width*0.5,height*0.5);
pop();
```

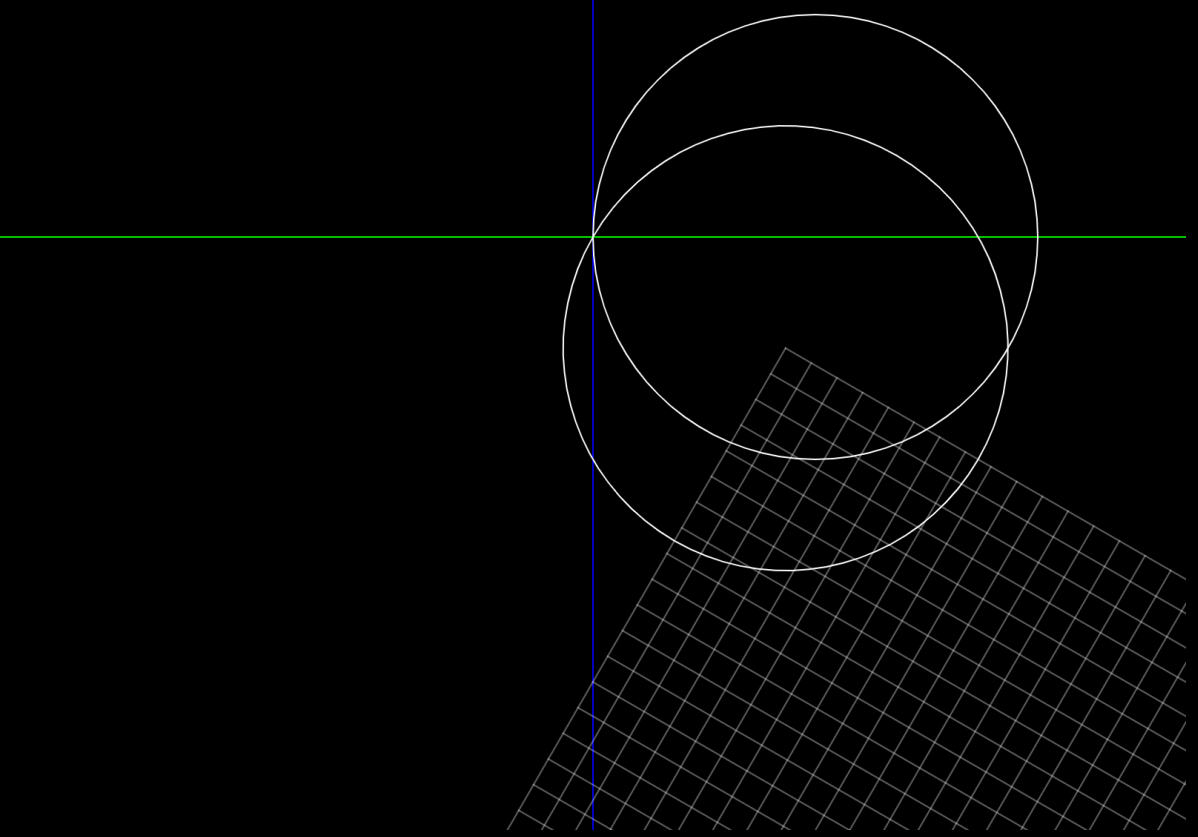


We need a rotated and translated twelve sets of circles to render this shape.

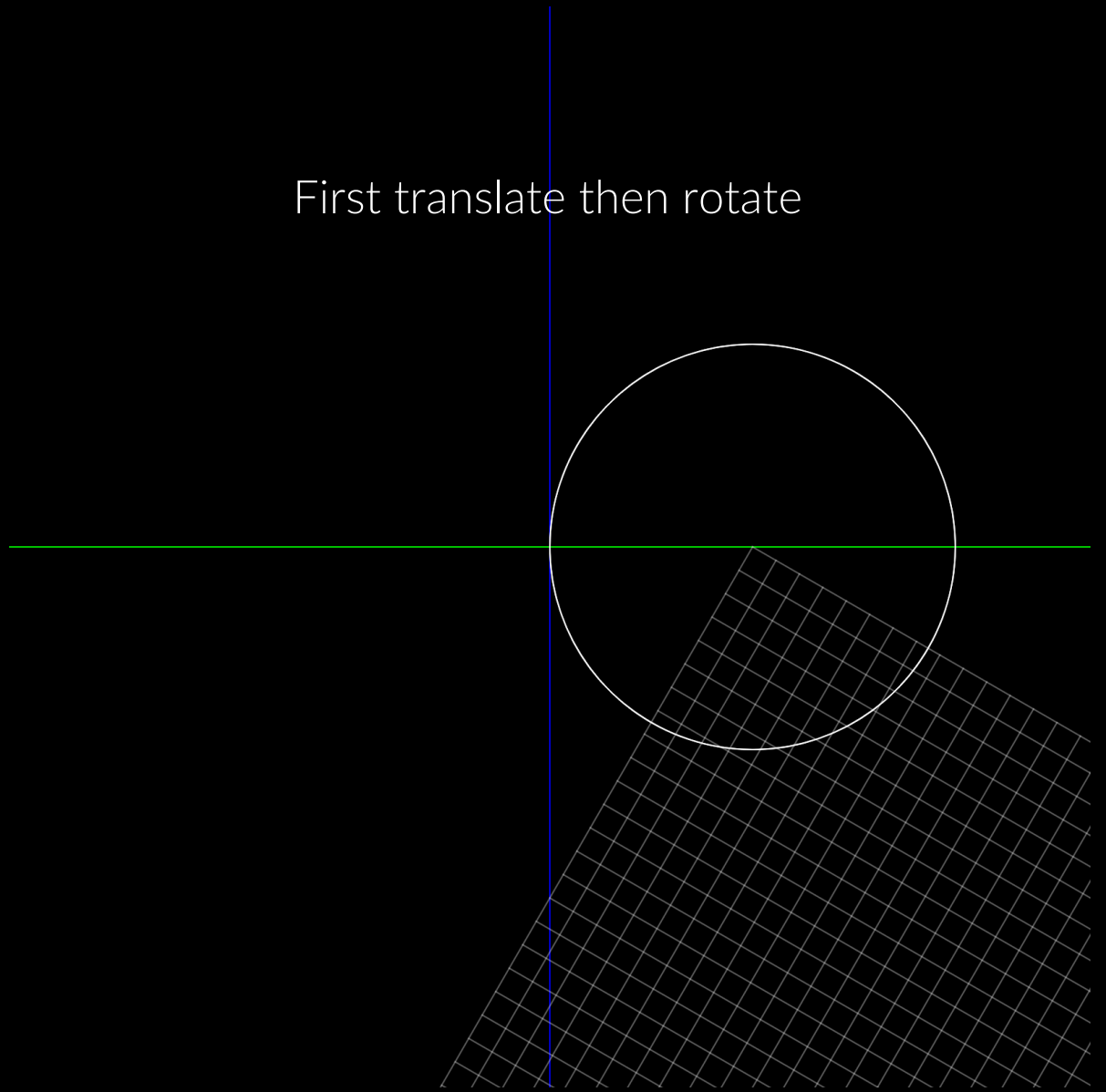
Algorithm: Rotate with multiples of 360/12=30 degrees each and translate the canvas to right with the distance of the radius of the circle.

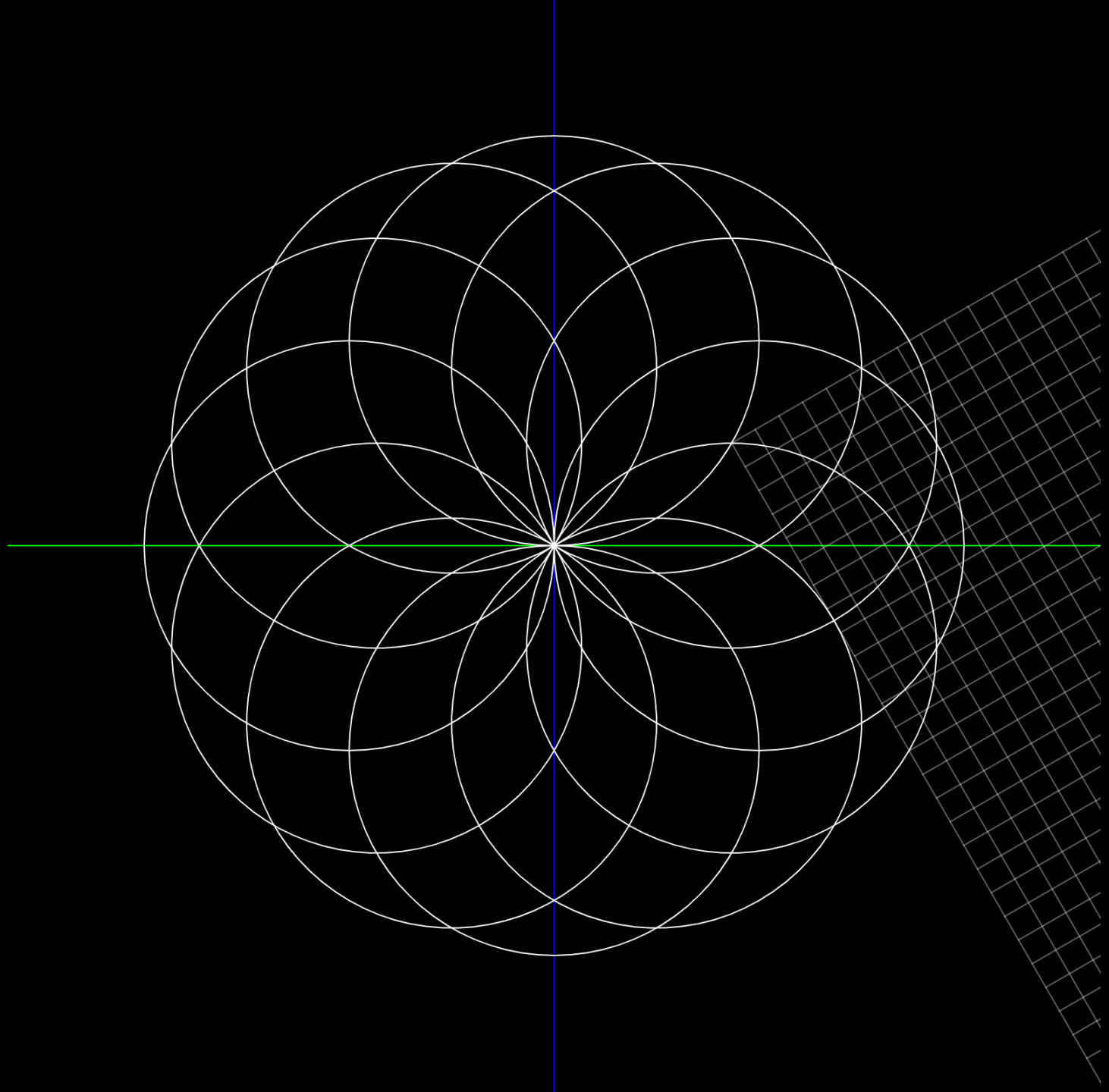
Transformation order is important!!!
Example below illustrating the second loop item

First rotate then translate

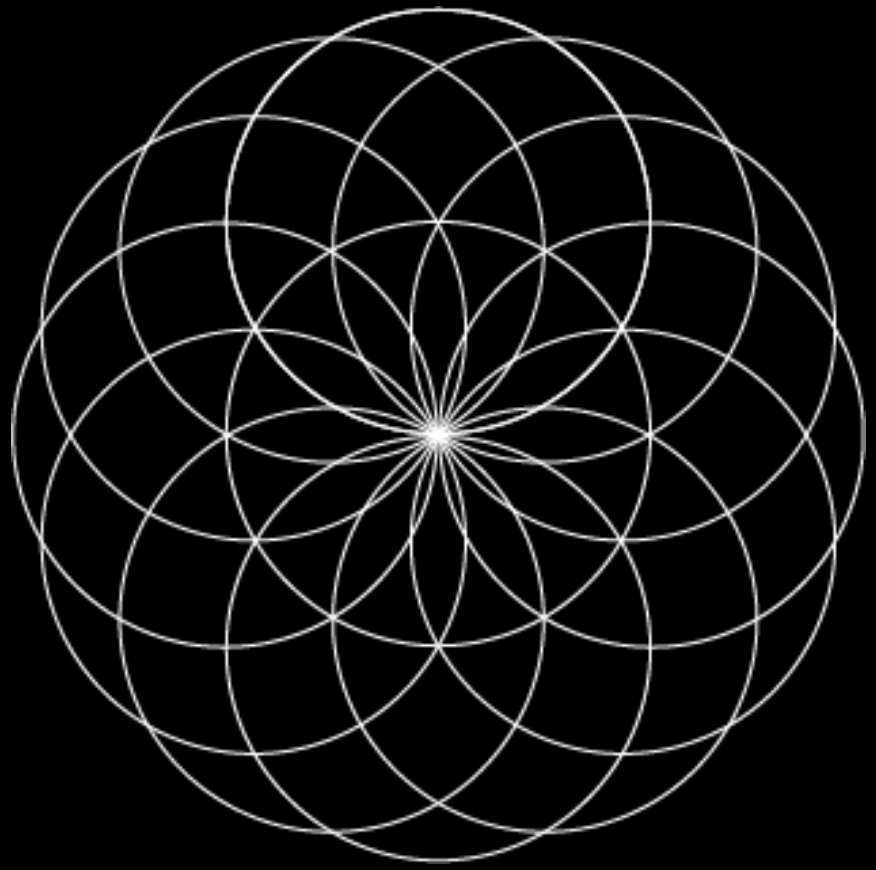


First translate then rotate





Apply twelve transformations



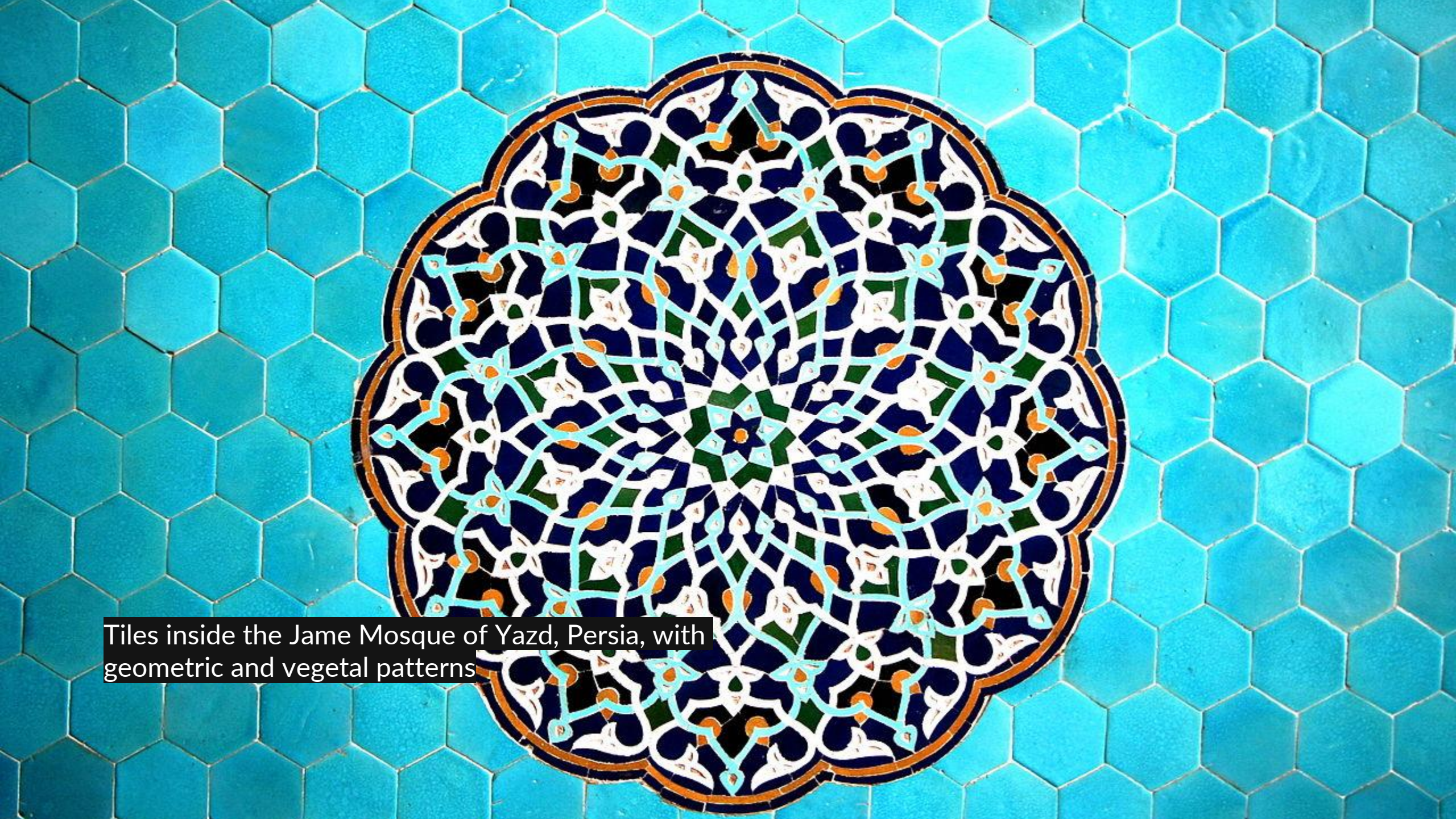
```
function setup() {
	createCanvas(400, 400);
	angleMode(DEGREES);
	noFill();
	stroke(255);
}

function draw() {
	background(0);
	push();
	translate(width*0.5,height*0.5);
	for(let i = 0; i < 12; i++){
		push();
		rotate(i*30);
		translate(50,0);
		circle(0,0,100);
		pop();
	}
	pop();
}
```

Islamic Geometric Patterns

The geometric designs in Islamic art are often built on combinations of repeated squares and circles, which may be overlapped and interlaced, as can arabesques (with which they are often combined), to form intricate and complex patterns, including a wide variety of tessellations.

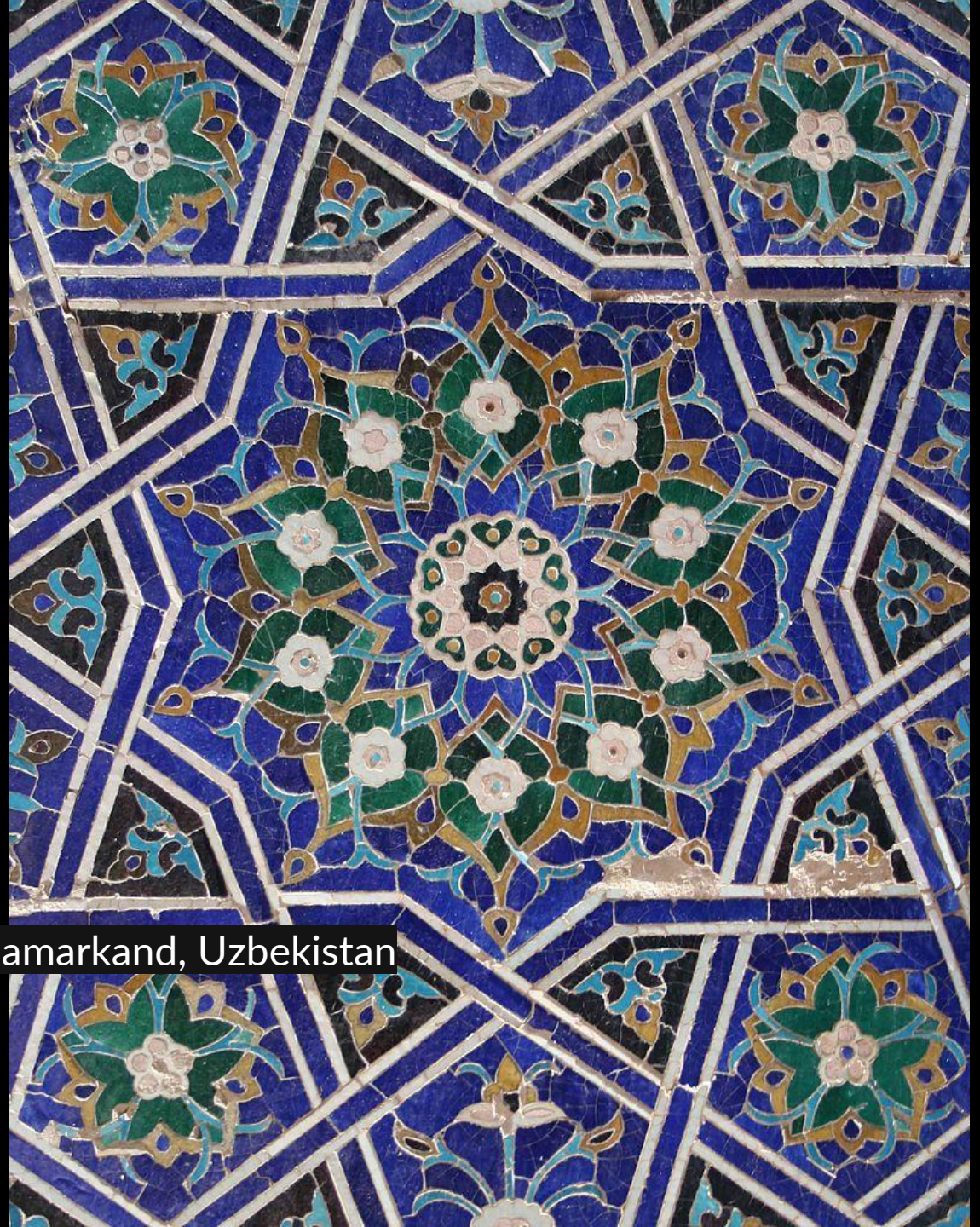
Islamic art mostly avoids figurative images to avoid becoming objects of worship. This aniconism in Islamic culture caused artists to explore non-figural art and created a general aesthetic shift toward mathematically-based decoration.

In Islamic culture, the patterns are believed to be the bridge to the spiritual realm, the instrument to purify the mind and the soul.

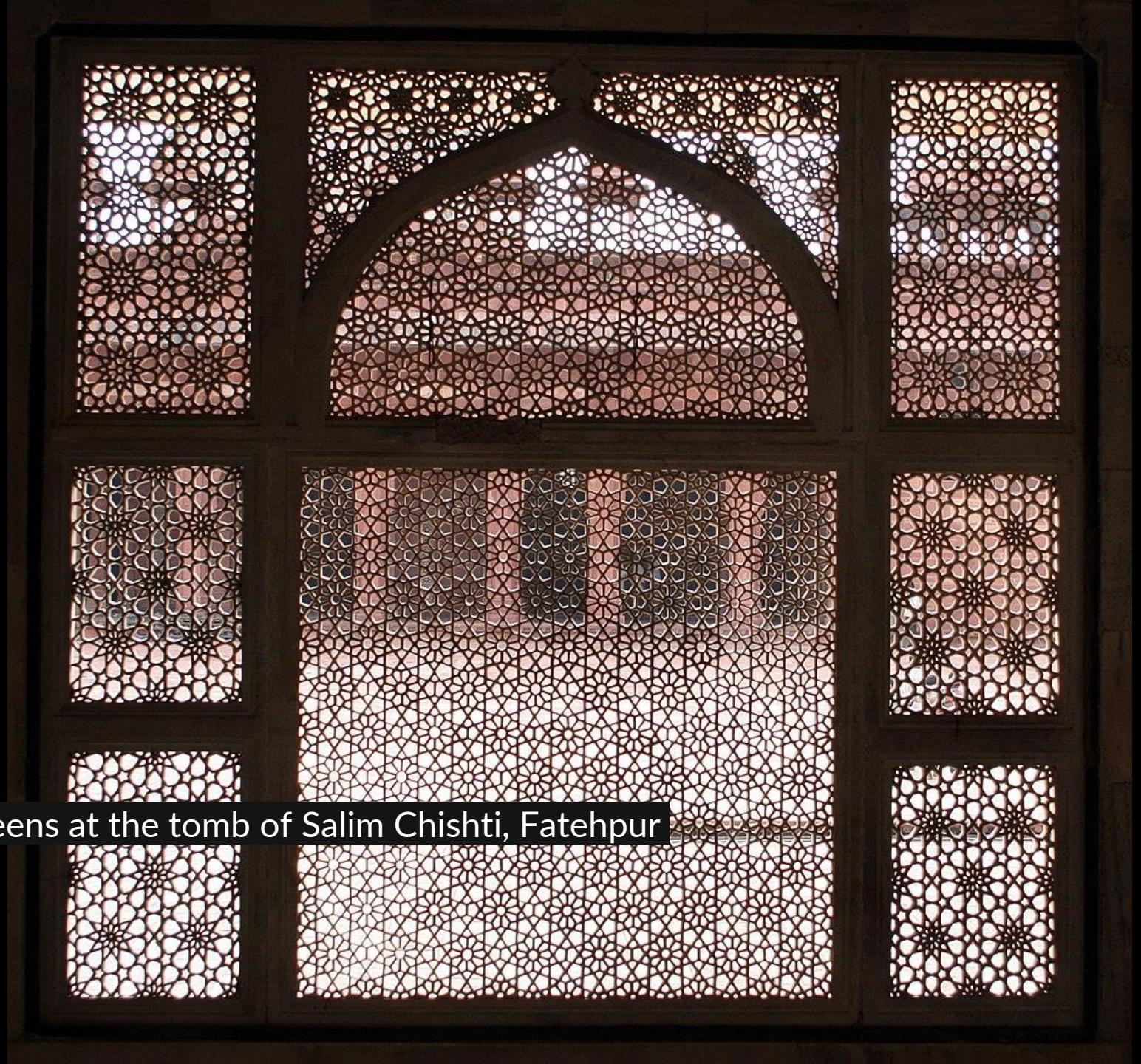


Tiles inside the Jame Mosque of Yazd, Persia, with geometric and vegetal patterns

The Shah Nematollah Vali Shrine, Mahan, Iran, 1431. The blue girih-tiled dome contains stars with, from the top, 5, 7, 9, 12, 11, 9 and 10 points in turn. 11-point stars are rare in Islamic art.[11]

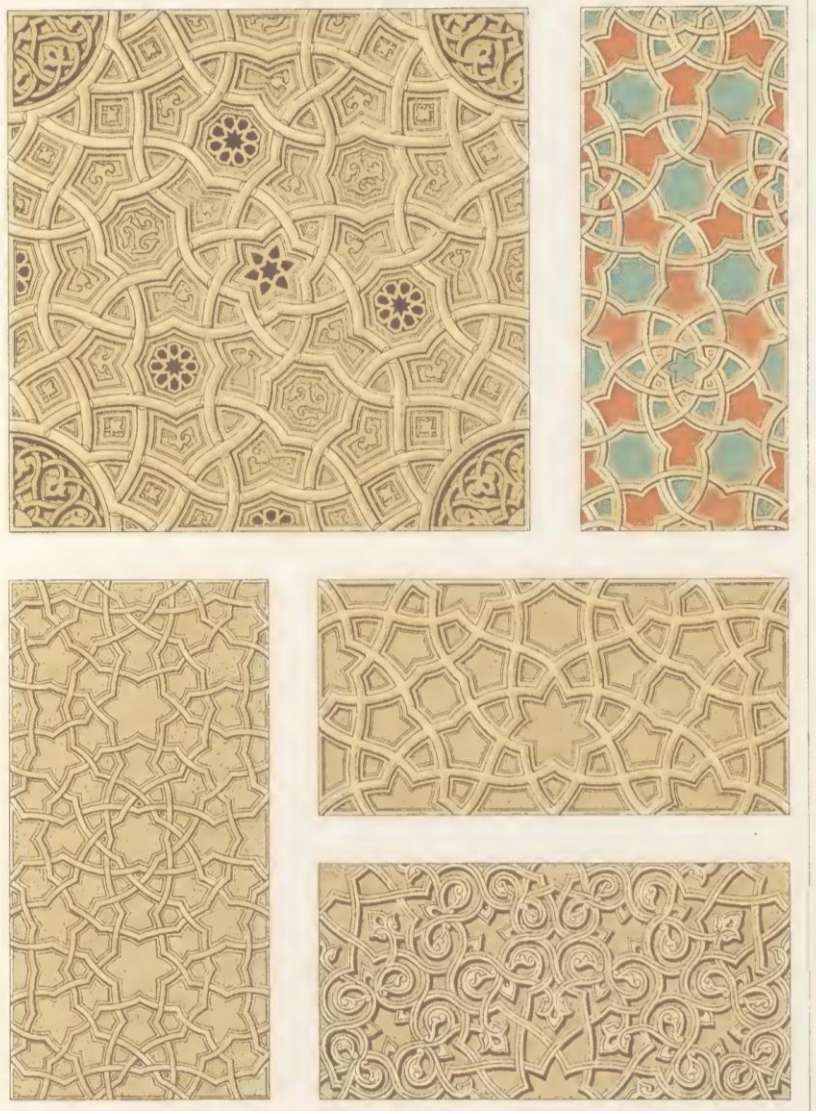


Girih at Shah-i-Zinda in Samarkand, Uzbekistan

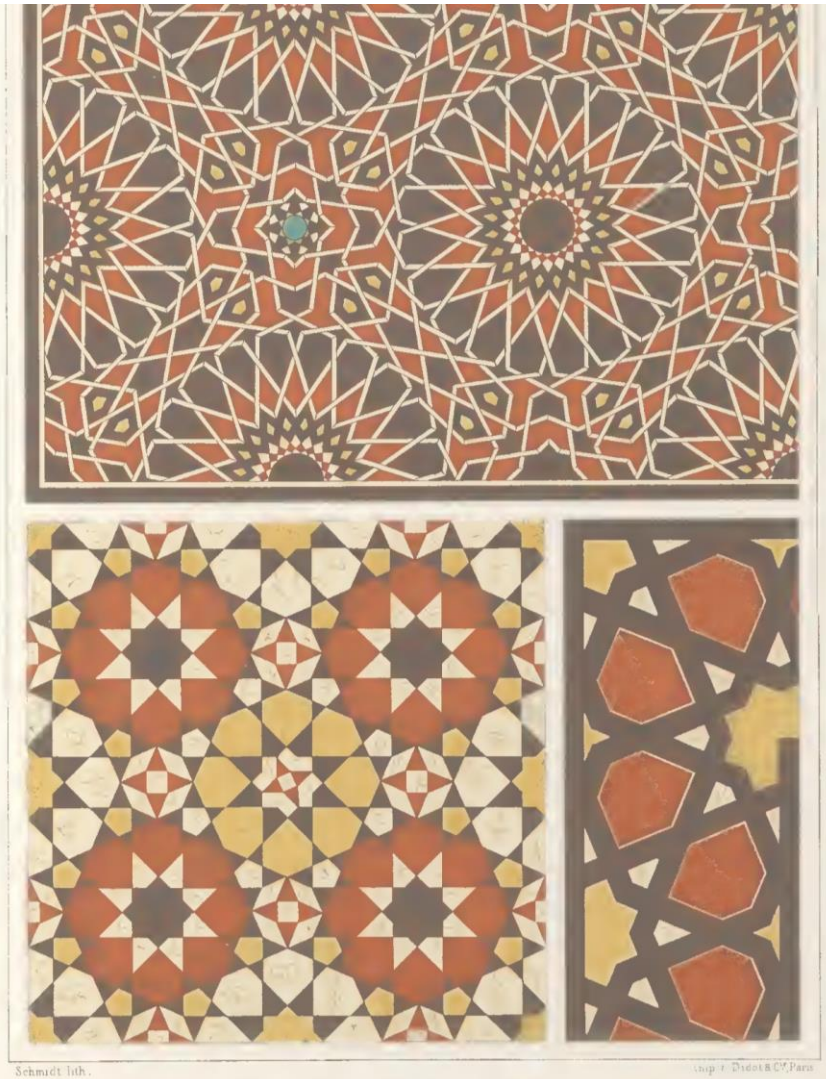


Jali pierced screens at the tomb of Salim Chishti, Fatehpur Sikri, India

PL. III.

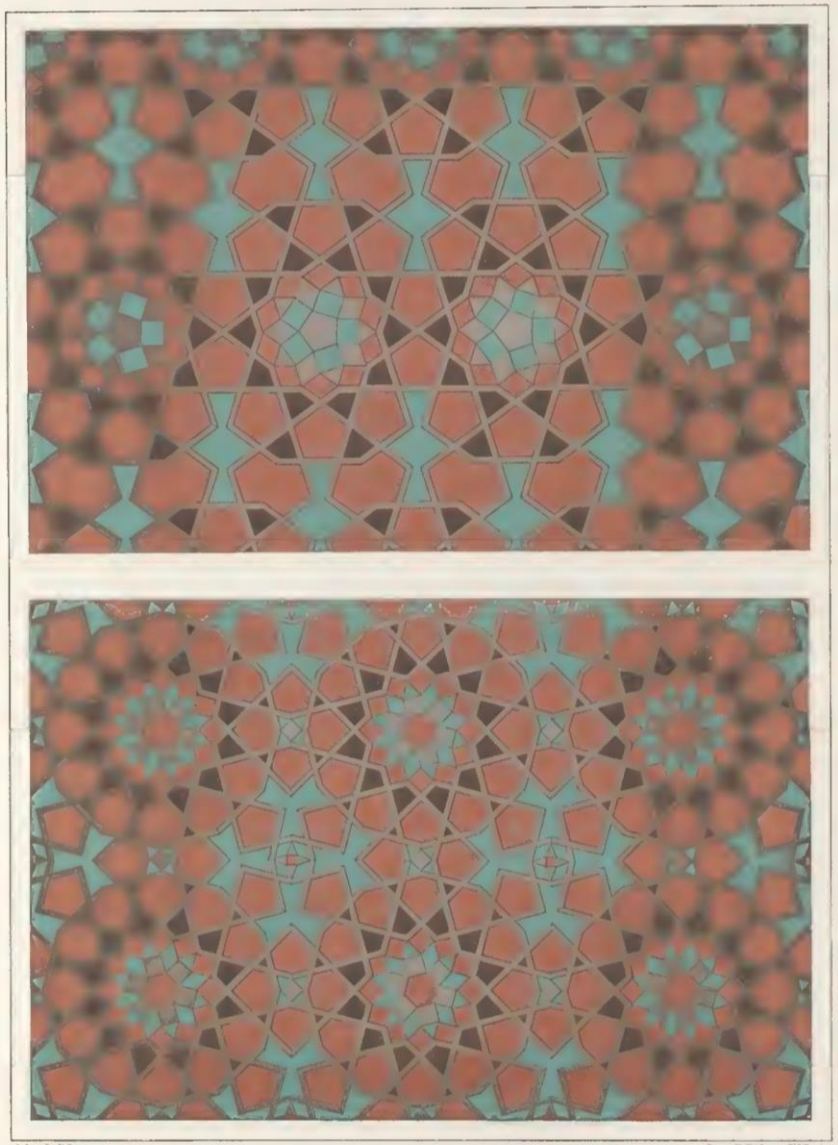


PL. II



Schmidt lith. Imp. F. Didot & Cie, Paris.

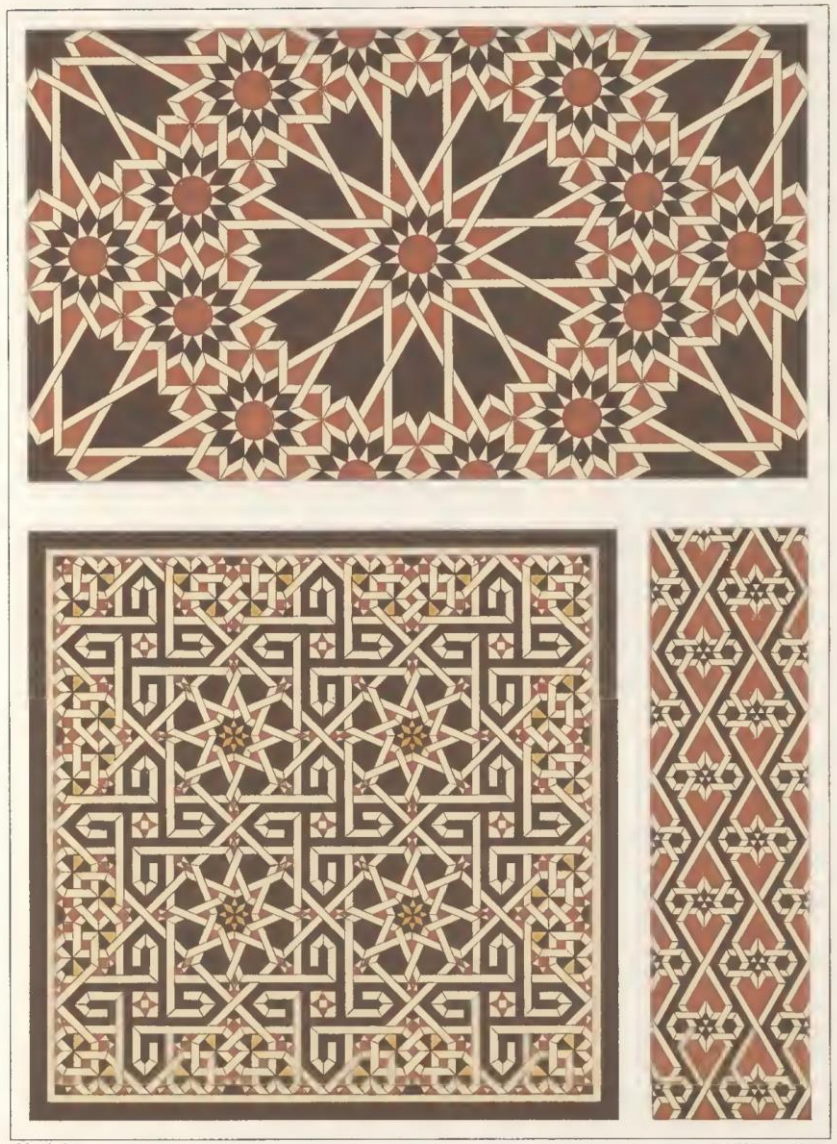
MOSAÏQUES



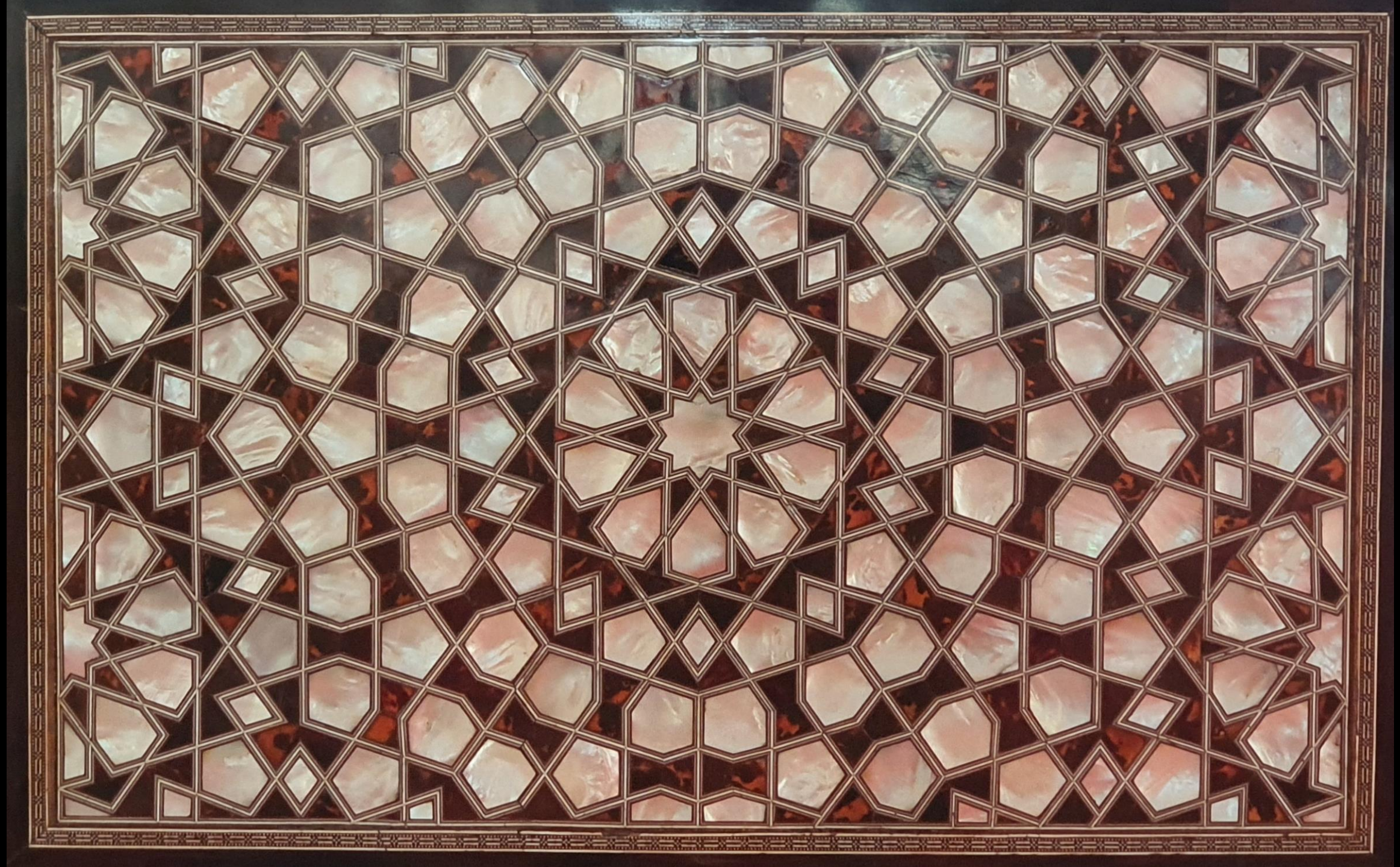
Schmidt lith. Imp. F. Didot & Cie, Paris.

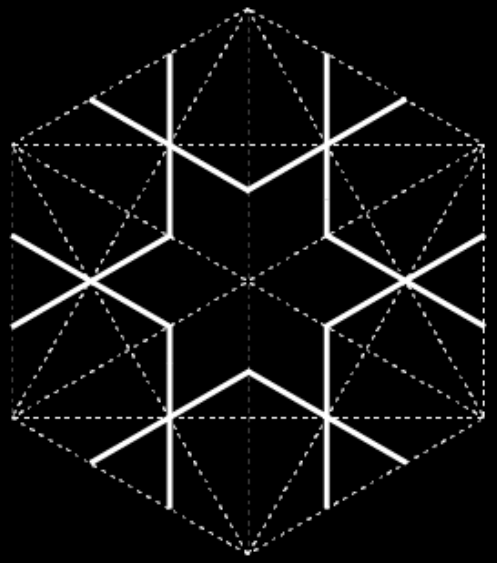
Schmidt, lith. Imp. F. Didot & Cie, Paris

INCRUSTATIONS

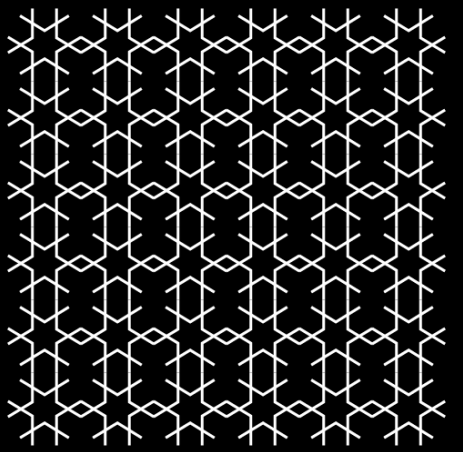


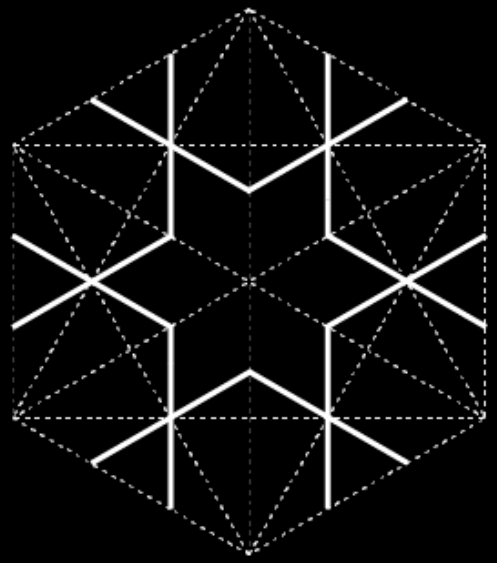
Schmidt, lith. Imp. F. Didot & Cie, Paris





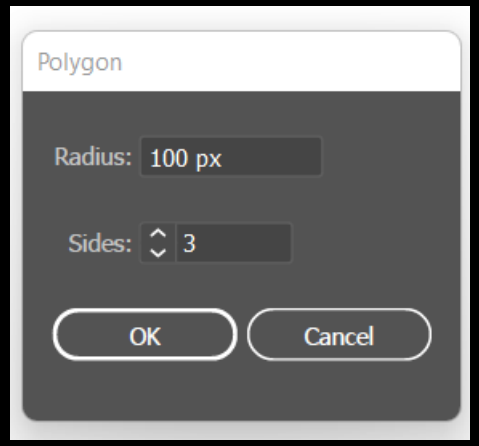
Ex: Coding an Islamic Geometric Pattern / Eşrefoğlu Mosque, Beyşehir

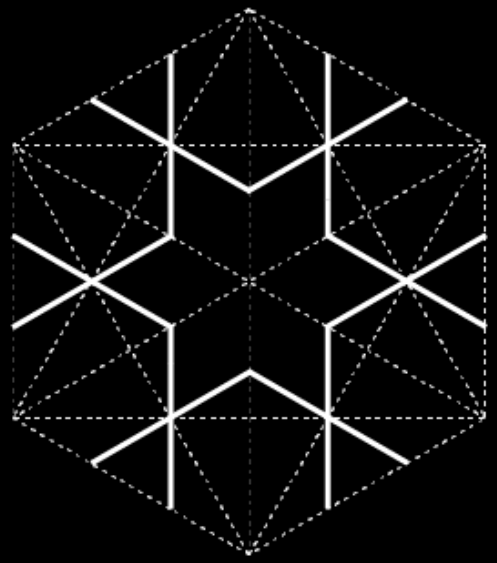




Illustrator Workflow

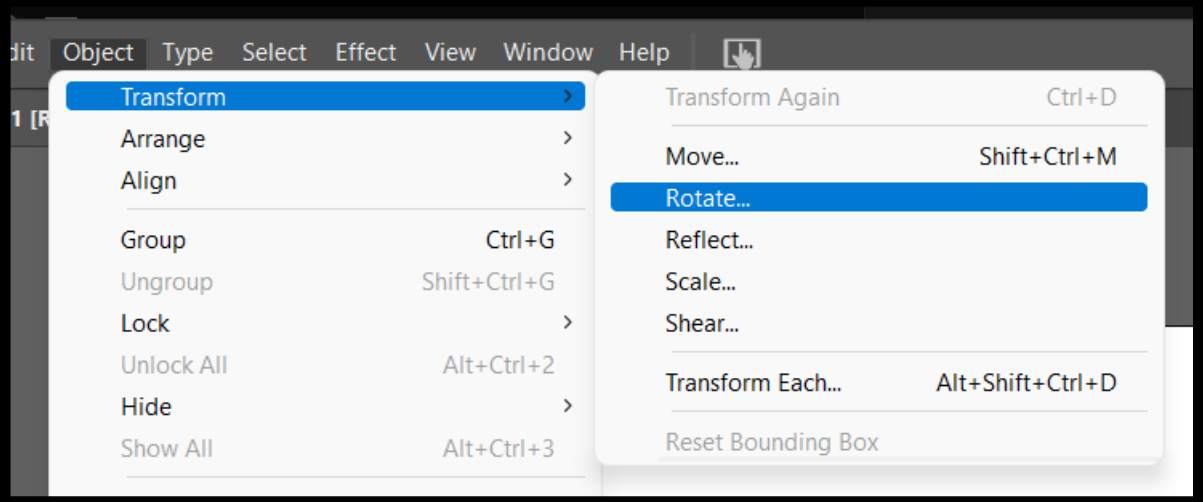
1-Create an artboard
2-Use Polygon tool to generate a triangle (no fill)

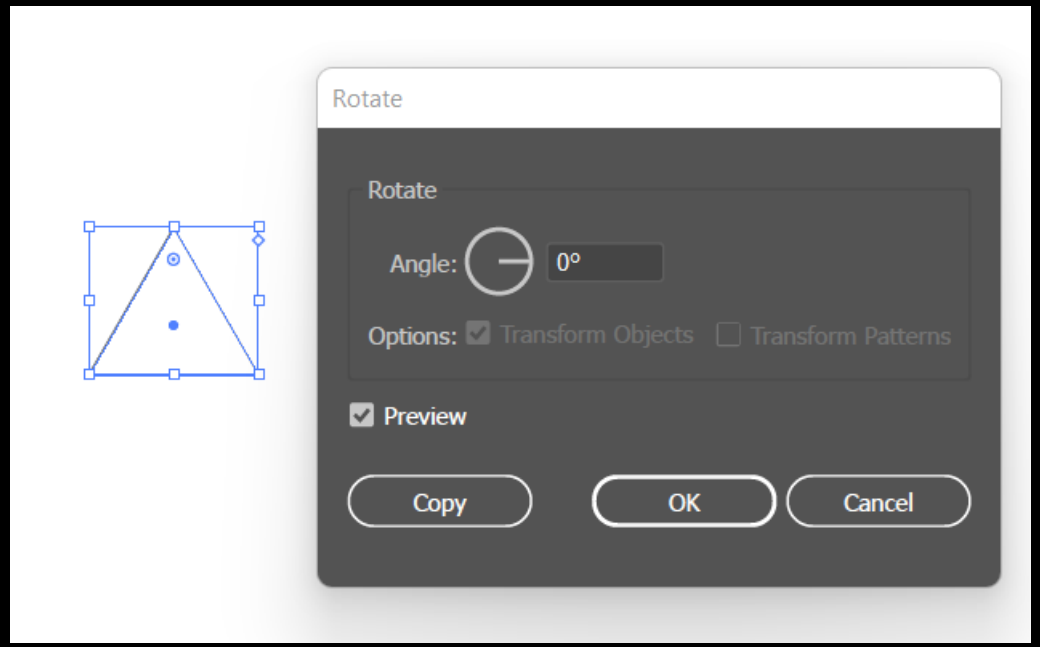


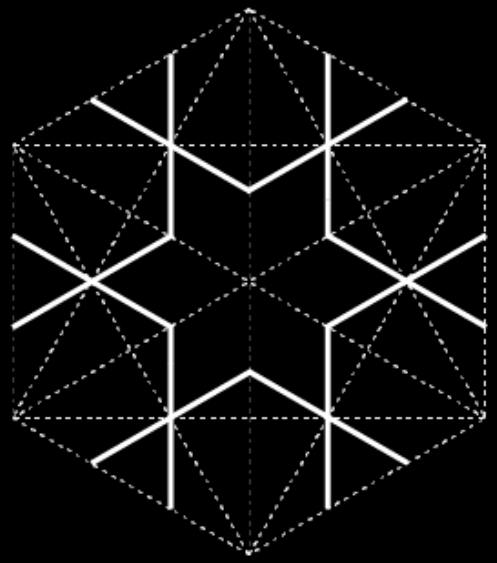


Illustrator Workflow

3-Use Object > Transform > Rotate (use angle 30 and Copy Option)

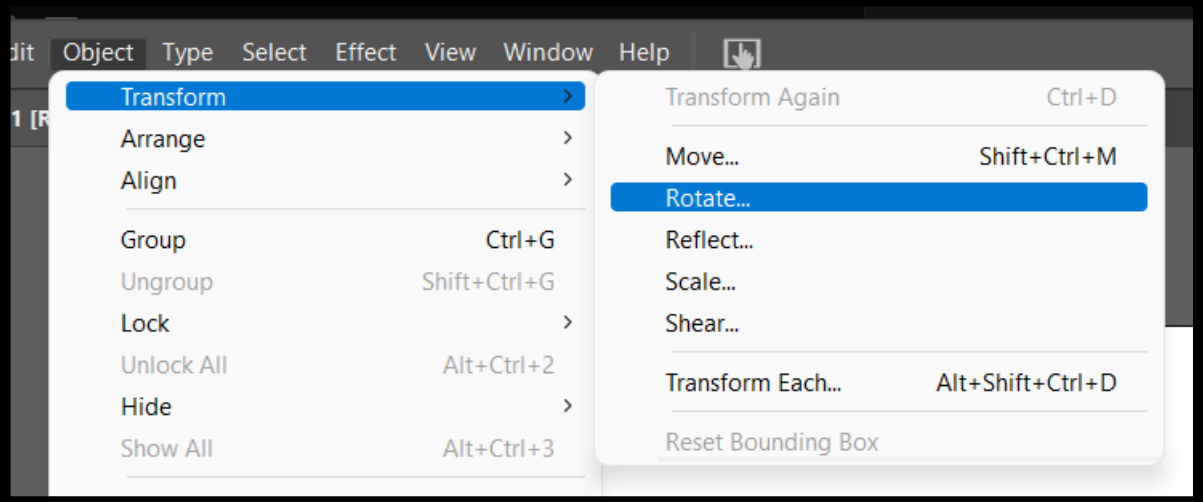


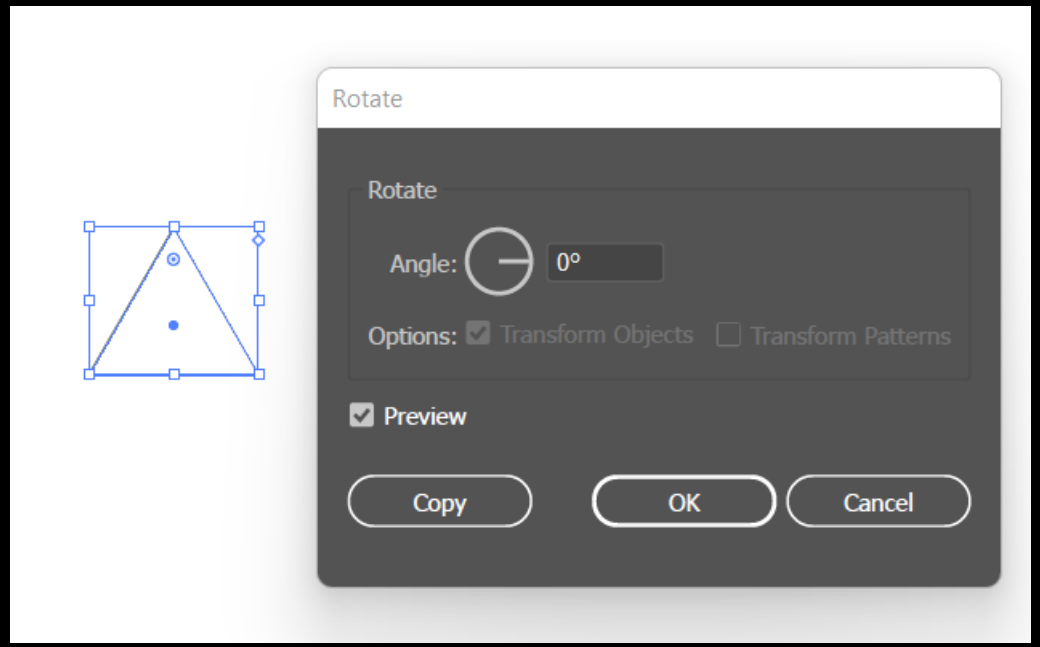


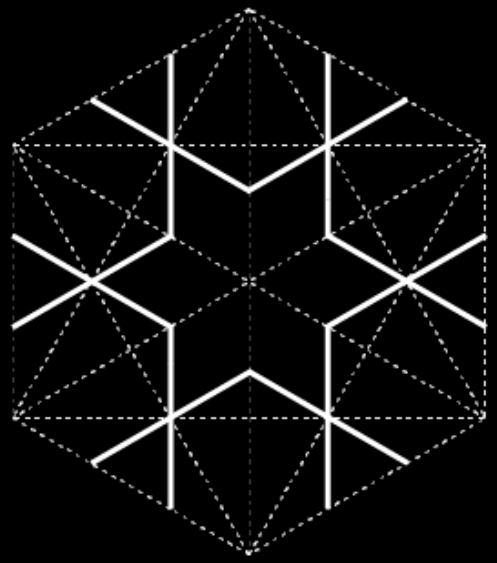


Illustrator Workflow

4-Use Object > Transform > Rotate (use angle -30 and Copy Option)

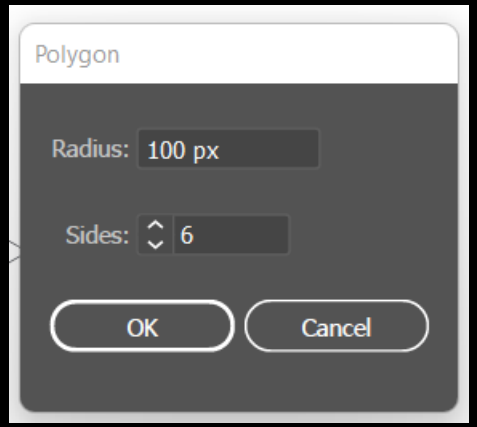


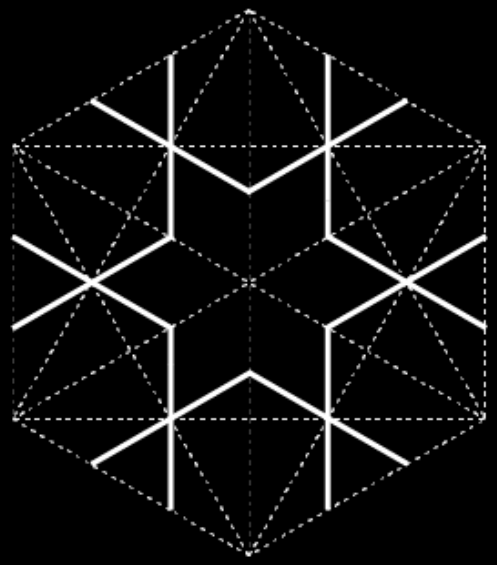




Illustrator Workflow

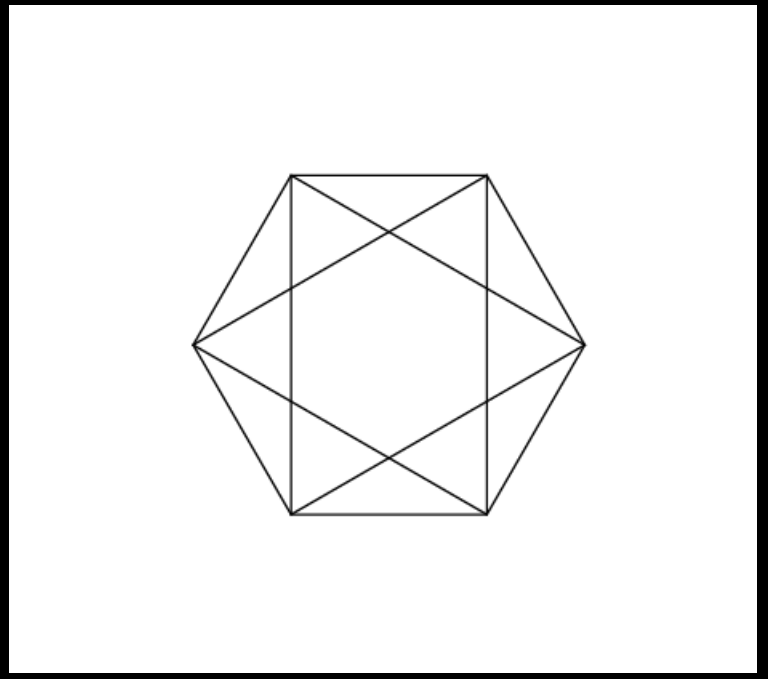
5-Use Polygon tool to generate a Hexagonal shape (no fill)

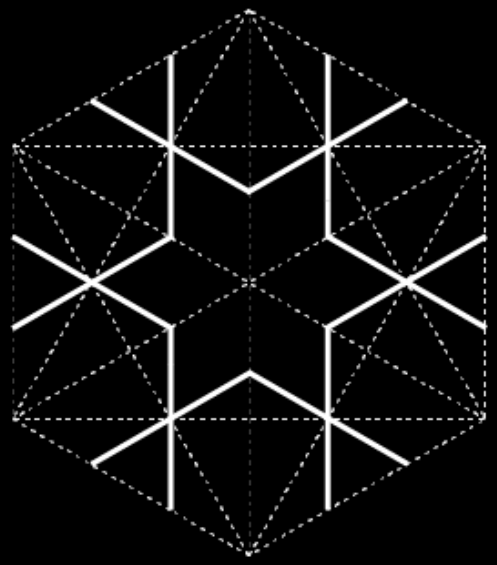




Illustrator Workflow

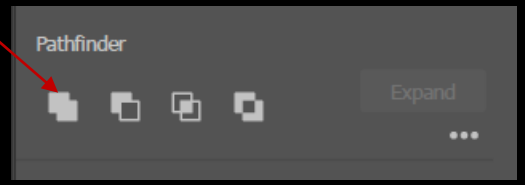
6-Place the two triangles in the Hexagonal shape so that they will be aligned accordingly

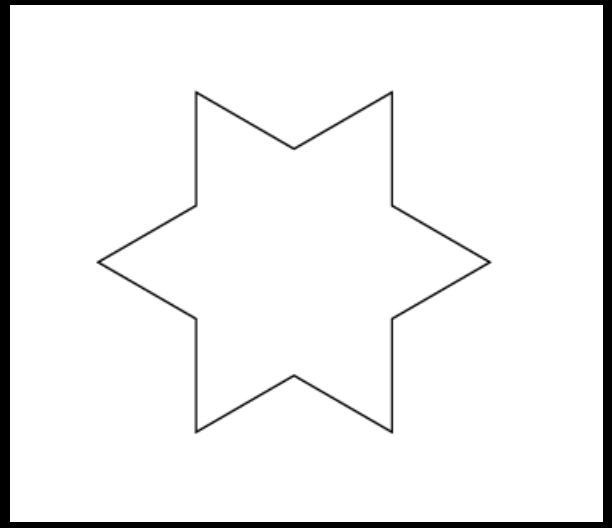


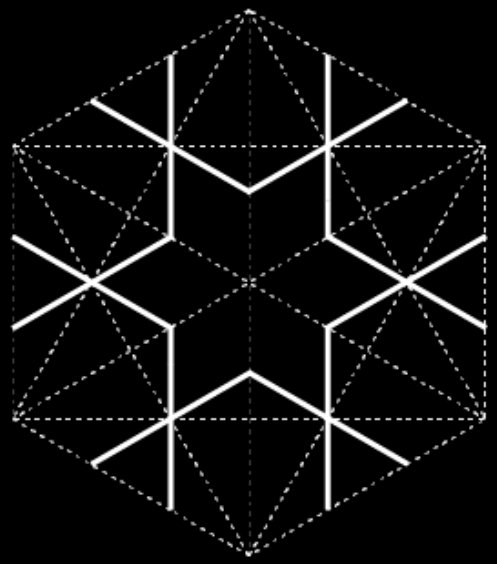


Illustrator Workflow

7-Delete the hexagon and choose two triangles and apply pathfinder unite function

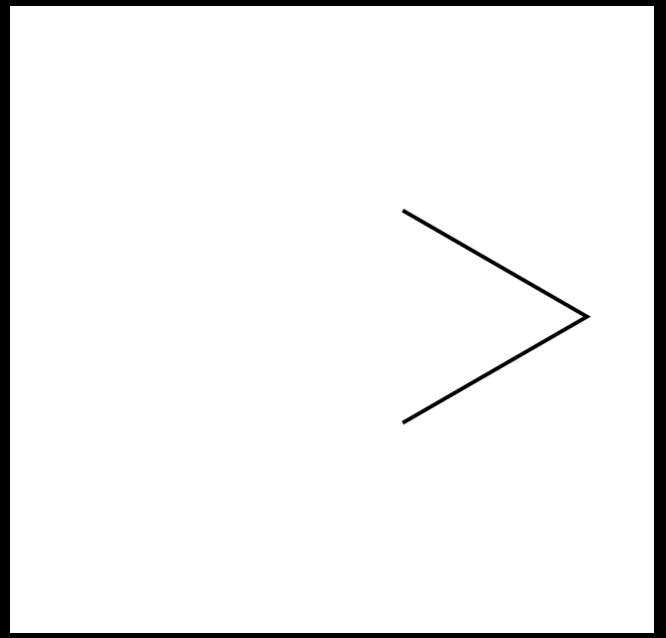


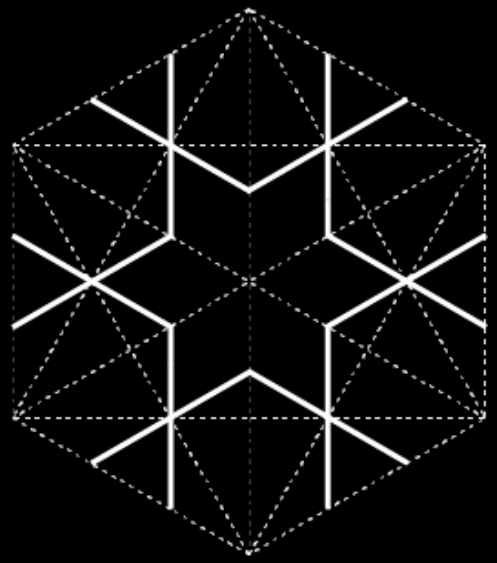




Illustrator Workflow

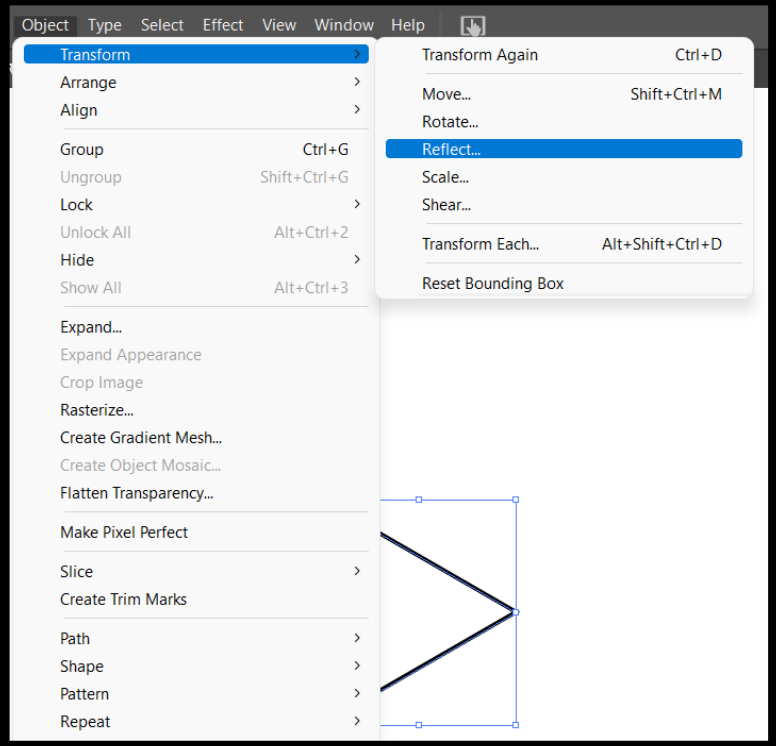
9-Use the direct selection tool to erase the five sides of the star.

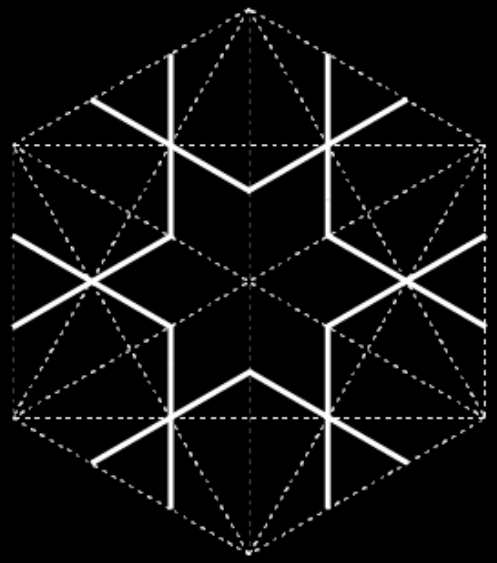




Illustrator Workflow

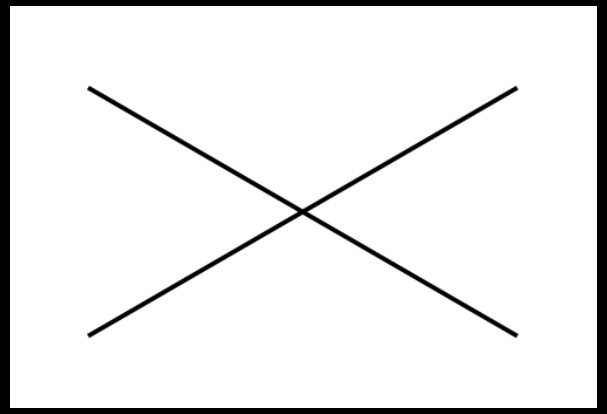
8-Object > Transform > Reflect (Use Vertical Copy Option)

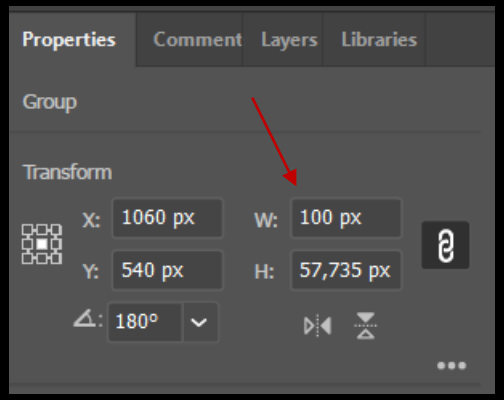


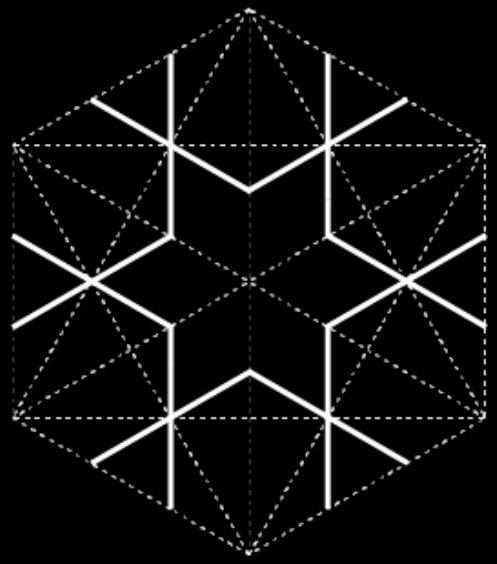


Illustrator Workflow

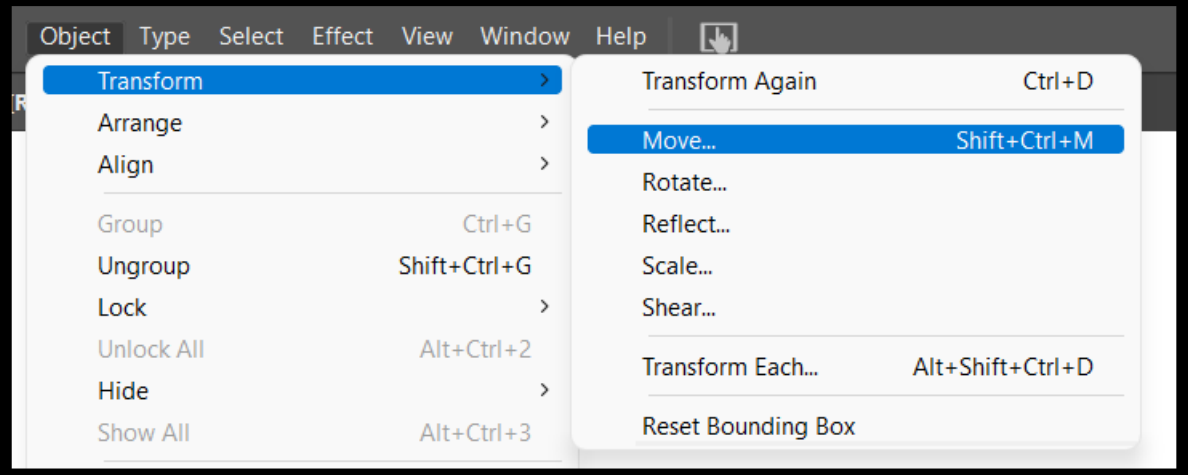
9- Move the reflected copy to the right and group (Ctrl+G) the arrowheads and we get our building block with 100 px wide

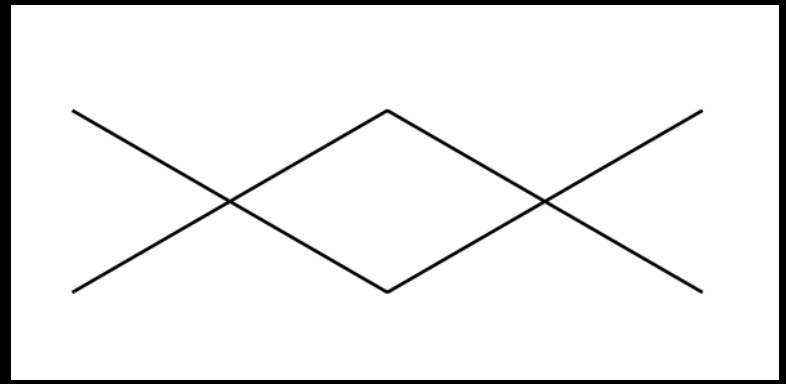


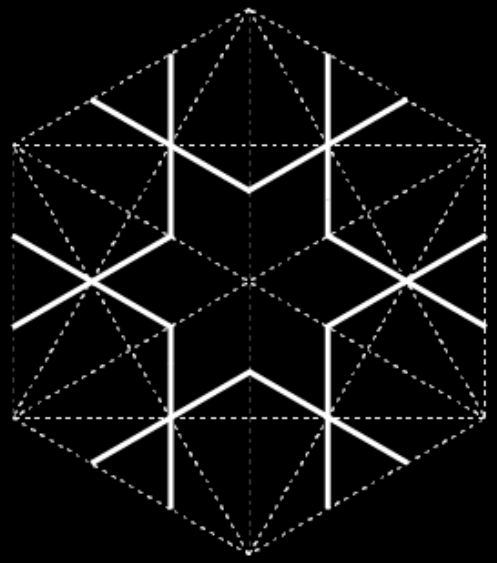




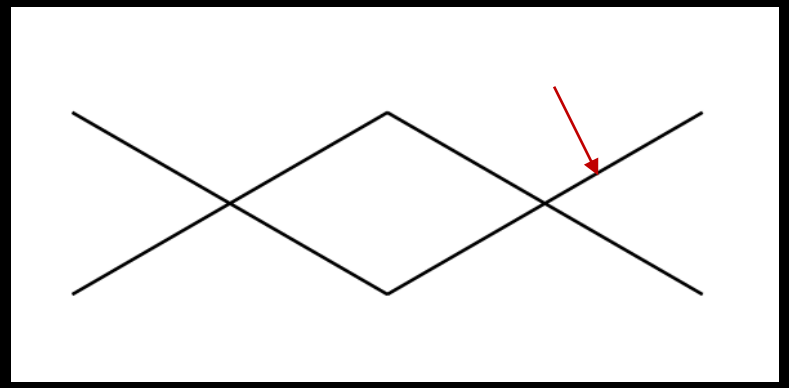
10- Object > Transform > Move the group shape 100 px right with the copy option

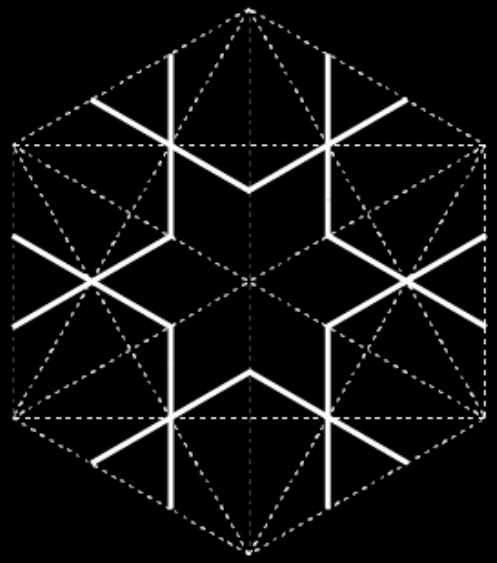




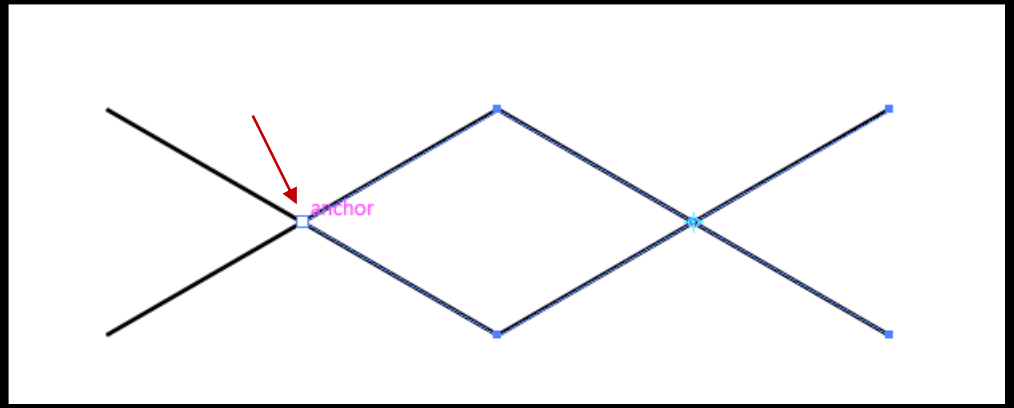


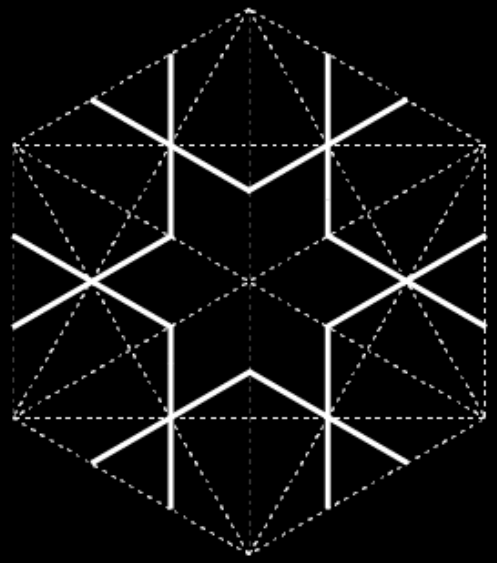
11- Choose the rightmost group and hit r on your keyboard to use the rotation tool





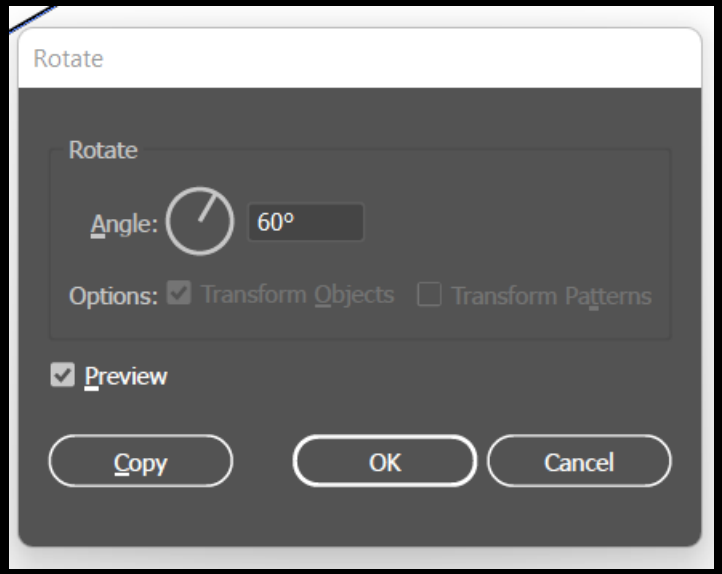
12- While holding the alt key choose the center of the leftmost group to position the anchor point of the rotation operation

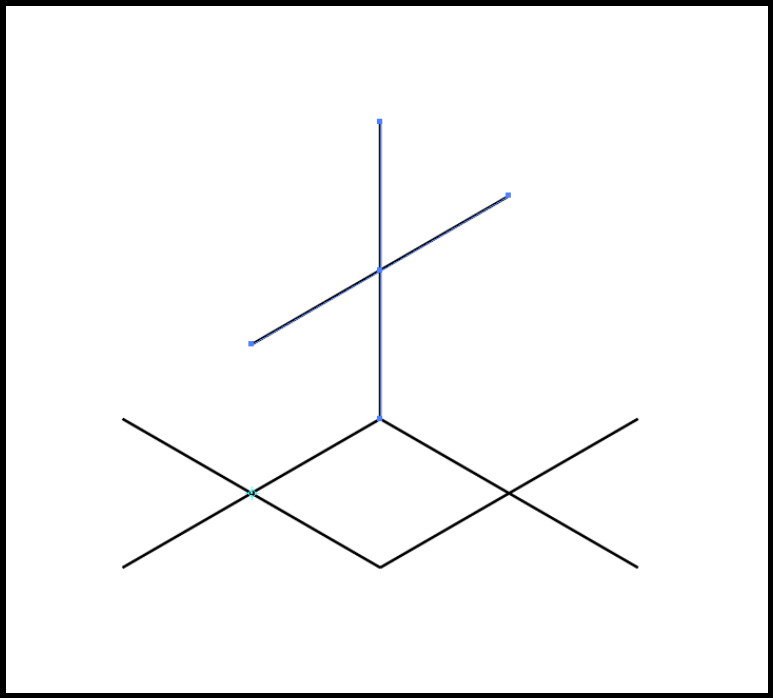


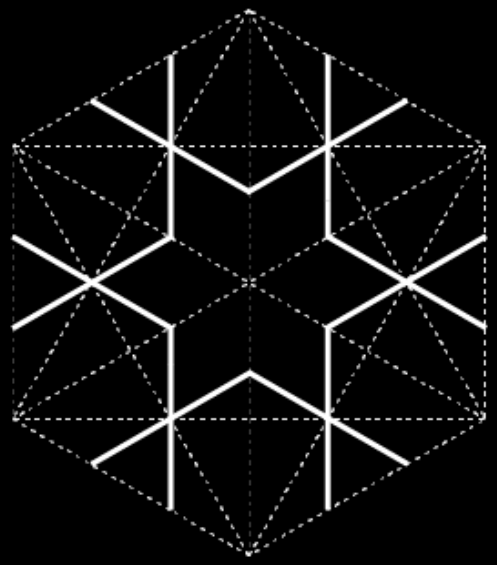


Illustrator Workflow

13- Apply the rotation with 60 degrees and copy option

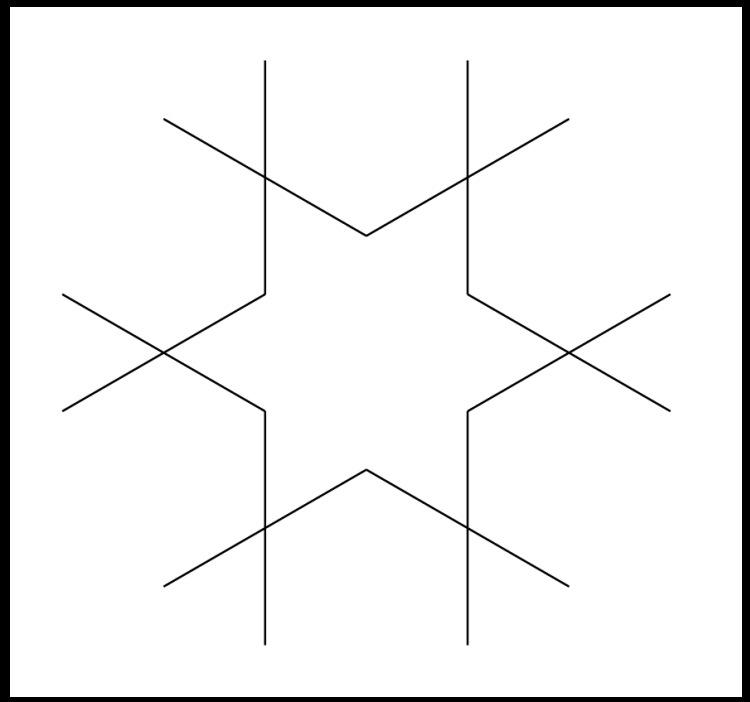






Illustrator Workflow

14- Repeat Ctrl + D to apply the rotation for the remaining sides and erase the center shape. Finale!

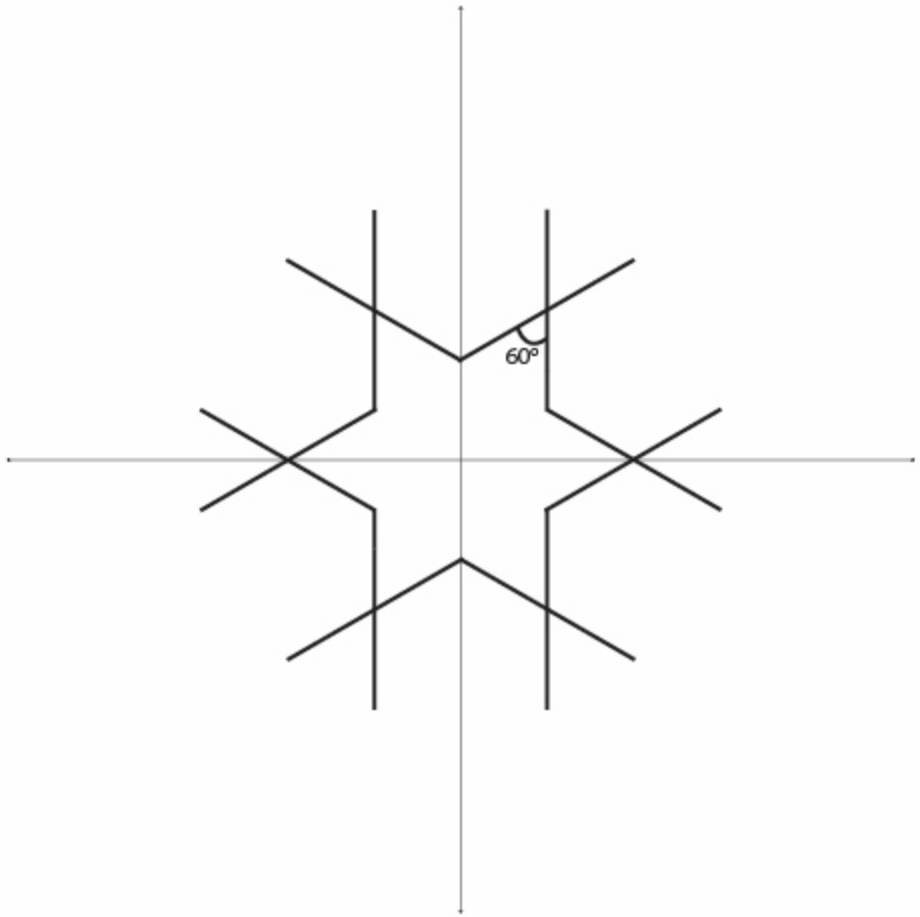


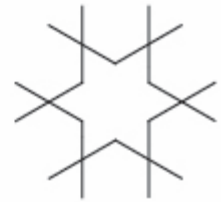
p5js Workflow

Coding an Islamic Geometric Pattern / Eşrefoğlu Mosque, Beyşehir

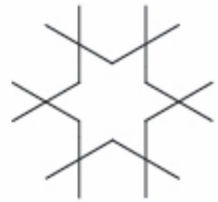
Selcuk ARTUT

Geometric Analysis





Coding an Islamic Geometric Pattern / Eşrefoğlu Mosque, Beyşehir
Selcuk ARTUT



Geometric Analysis

Finding the vertex points

$x_0 = -1 * \cos(30) * r$

$y_0 = -1 * \sin(30) * r$

$x_1 = \cos(30) * r$

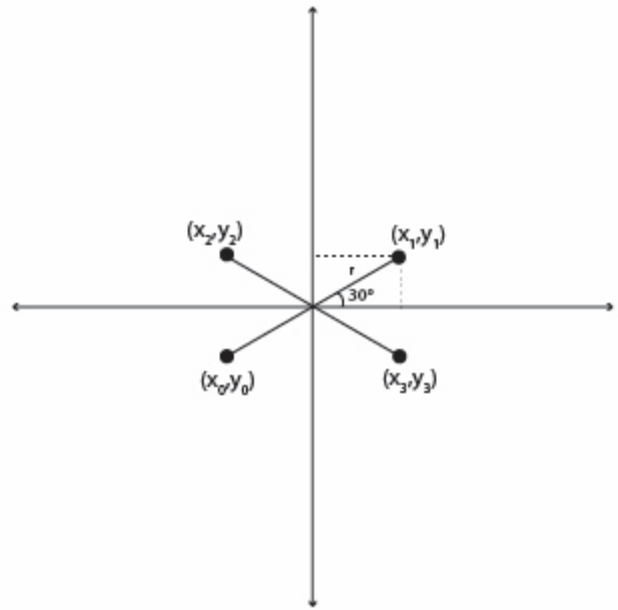
$y_1 = \sin(30) * r$

$x_2 = -1 * \cos(30) * r$

$y_2 = \sin(30) * r$

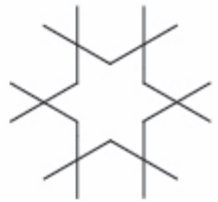
$x_3 = \cos(30) * r$

$y_3 = -1 * \sin(30) * r$



Coding an Islamic Geometric Pattern / Eşrefoğlu Mosque, Beyşehir

Selcuk ARTUT



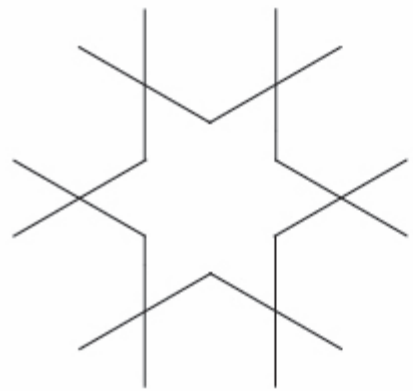
Code

```
let r = 50;

function setup() {
  createCanvas(400, 400);
  angleMode(DEGREES);
  stroke(0);
  noLoop();
}

function draw() {
  background(255);
  push();
  translate(width*0.5,height*0.5);
  beginShape();
  let x0 = -1 * cos(30) * r;
  let y0 = -1 * sin(30) * r;
  vertex(x0,-y0);
  let x1 = cos(30) * r;
  let y1 = sin(30) * r;
  vertex(x1,-y1);
  endShape();
  beginShape();
  let x2 = -1 * cos(30) * r;
  let y2 = sin(30) * r;
  vertex(x2,-y2);
  let x3 = cos(30) * r;
  let y3 = -1 * sin(30) * r;
  vertex(x3,-y3);
  endShape();
  pop();
}
```

Coding an Islamic Geometric Pattern / Eşrefoğlu Mosque, Beyşehir
Selcuk ARTUT

Code - Copy and Transformation

```
let r = 50;

function setup() {
  createCanvas(400, 400);
  angleMode(DEGREES);
  stroke(0);
  noLoop();
}

function draw() {
  background(255);
  push();
  translate(width * 0.5, height * 0.5);
  for (let i = 0; i < 6; i++) {
    push();
    rotate(i*60);
    translate(cos(30) * r * 2, 0);
    beginShape();
    let x0 = -1 * cos(30) * r;
    let y0 = -1 * sin(30) * r;
    vertex(x0, -y0);
    let x1 = cos(30) * r;
    let y1 = sin(30) * r;
    vertex(x1, -y1);
    endShape();
    beginShape();
    let x2 = -1 * cos(30) * r;
    let y2 = sin(30) * r;
    vertex(x2, -y2);
    let x3 = cos(30) * r;
    let y3 = -1 * sin(30) * r;
    vertex(x3, -y3);
    endShape();
    pop();
  }
  pop();
}
```



Exporting SVG models from p5.js

There is an external library to export SVG files from the scripts.

It has to be included in the index.html as below:

```
<html>
<head>
  <meta charset="UTF-8">
  <script language="javascript" type="text/javascript" src="libraries/p5.js"></script>
  <script language="javascript" type="text/javascript" src="sketch.js"></script>
  <script src="https://unpkg.com/p5.js-svg@1.1.1"></script>
</head>

<body>
</body>
</html>
```

https://unpkg.com/p5.js-svg@1.1.1

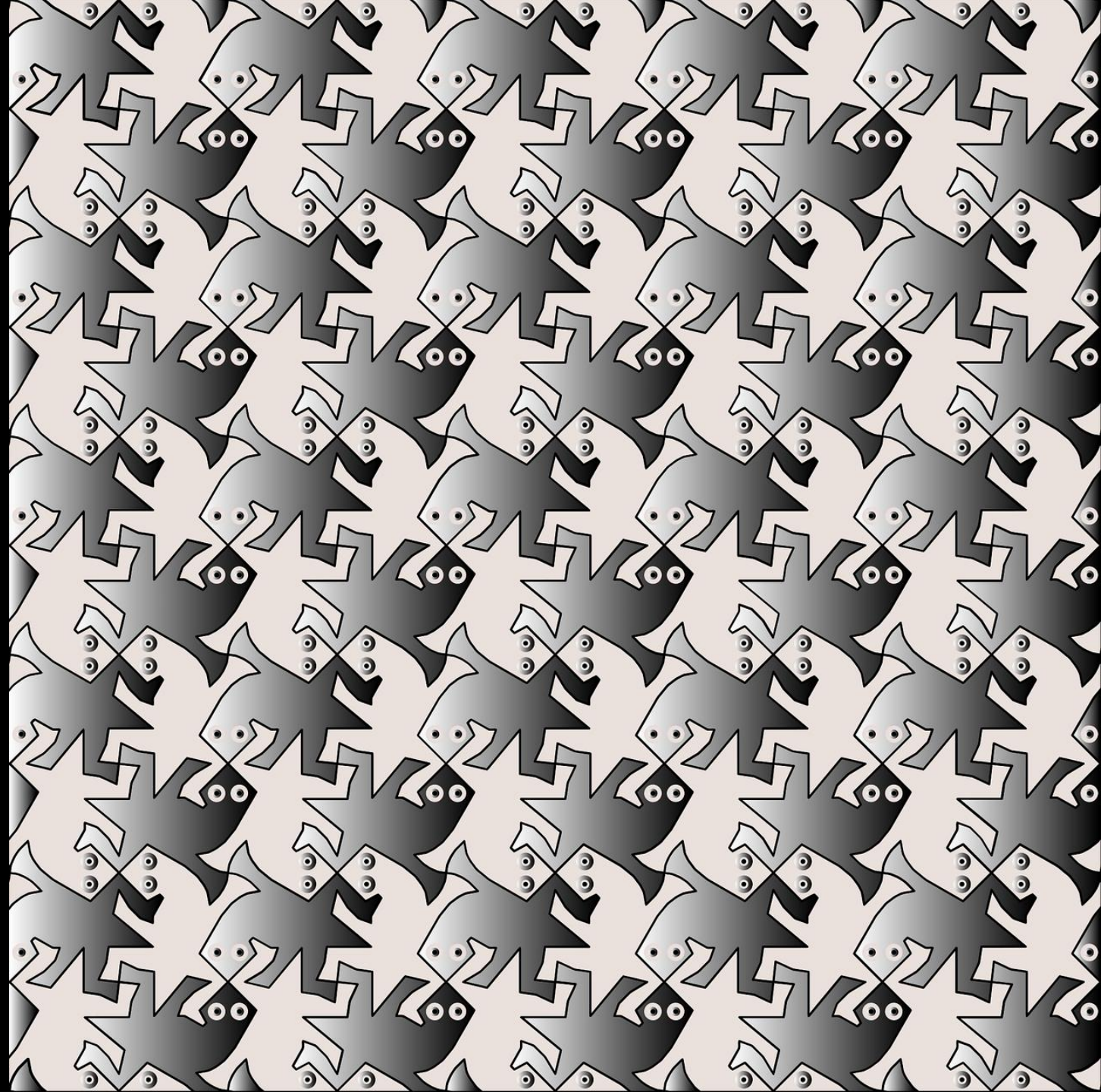
Exporting SVG models from p5.js

Change sketch.js as below;

```
function setup() {
        createCanvas(900, 600, SVG);
}

function draw() {
    // draw your visual here
    save("mySVG.svg"); // give file name
    noLoop();
}
```

Tesselation

A tessellation or tiling is the covering of a surface, often a plane, using one or more geometric shapes, called tiles, with no overlaps and no gaps. In mathematics, tessellation can be generalized to higher dimensions and a variety of geometries.



Tesselation in p5.js

```
class Rectangle {
  constructor(name, height, width) {
    this.name = name;
    this.height = height;
    this.width = width;
  }
}

let square = new Rectangle('square', 1, 1); // creating new
instance of Polygon Class.
console.log(square.width); // prints '1' to the console
```

Tesselation in p5.js

```
let bug; // Declare object
function setup() {
	createCanvas(710, 400);  // Create object
	bug = new Jitter();
}
function draw() {
	background(50, 89, 100);
	bug.move();
	bug.display();
}
```

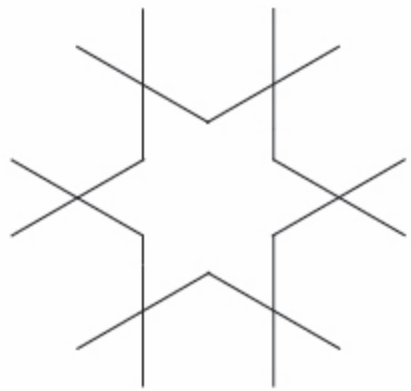
```
// Jitter class
class Jitter {
	constructor() {
		this.x = random(width);
		this.y = random(height);
		this.diameter = random(10, 30);
		this.speed = 1;
	}

	move() {
		this.x += random(-this.speed, this.speed);
		this.y += random(-this.speed, this.speed);
	}

	display() {
		ellipse(this.x, this.y, this.diameter, this.diameter);
	}
}
```

Coding an Islamic Geometric Pattern / Eşrefoğlu Mosque, Beyşehir
Selcuk ARTUT

Code - Tesselation

Lets turn the motif code into a class, so that we may place the motif in multiple times

```
// Motif class
class Motif {
 constructor(r) {
  this.r = r;
 }

 display() {
  for (let i = 0; i < 6; i++) {
   push();
   rotate(i*60);
   translate(cos(30) * this.r * 2, 0);
   beginShape();
   let x0 = -1 * cos(30) * this.r;
   let y0 = -1 * sin(30) * this.r;
   vertex(x0, -y0);
   let x1 = cos(30) * this.r;
   let y1 = sin(30) * this.r;
   vertex(x1, -y1);
   endShape();
   beginShape();
   let x2 = -1 * cos(30) * this.r;
   let y2 = sin(30) * this.r;
   vertex(x2, -y2);
   let x3 = cos(30) * this.r;
   let y3 = -1 * sin(30) * this.r;
   vertex(x3, -y3);
   endShape();
   pop();
  }
 }
}
```

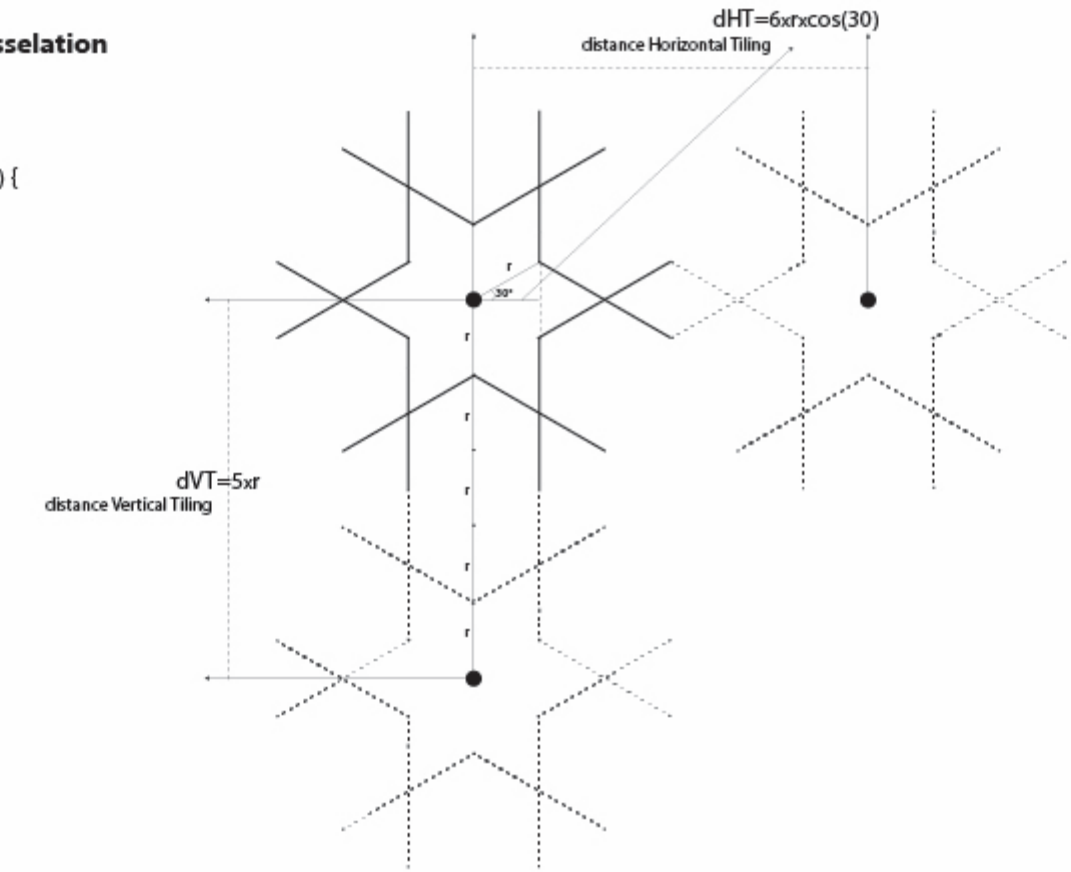


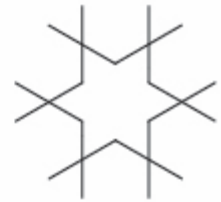
Coding an Islamic Geometric Pattern / Eşrefoğlu Mosque, Beyşehir

Selcuk ARTUT

Code - Tesselation

```
// Tile class
class Tile {
 constructor() {

 }

 display() {

 }
}
```



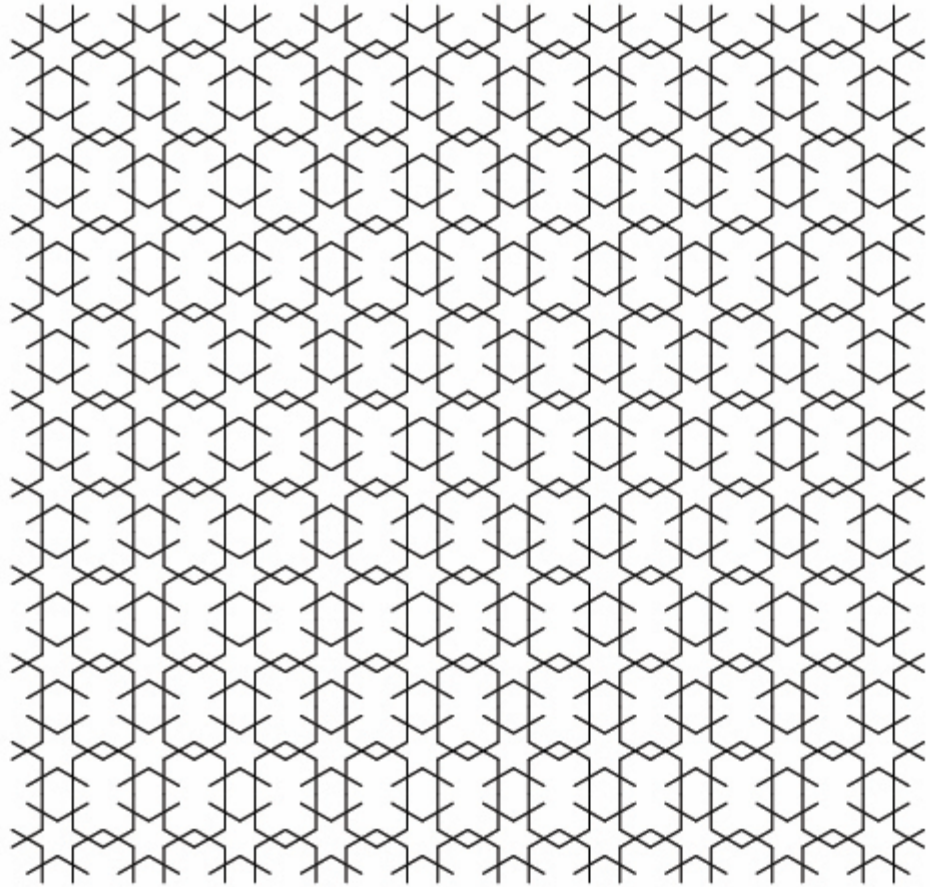


Coding an Islamic Geometric Pattern / Eşrefoğlu Mosque, Beyşehir

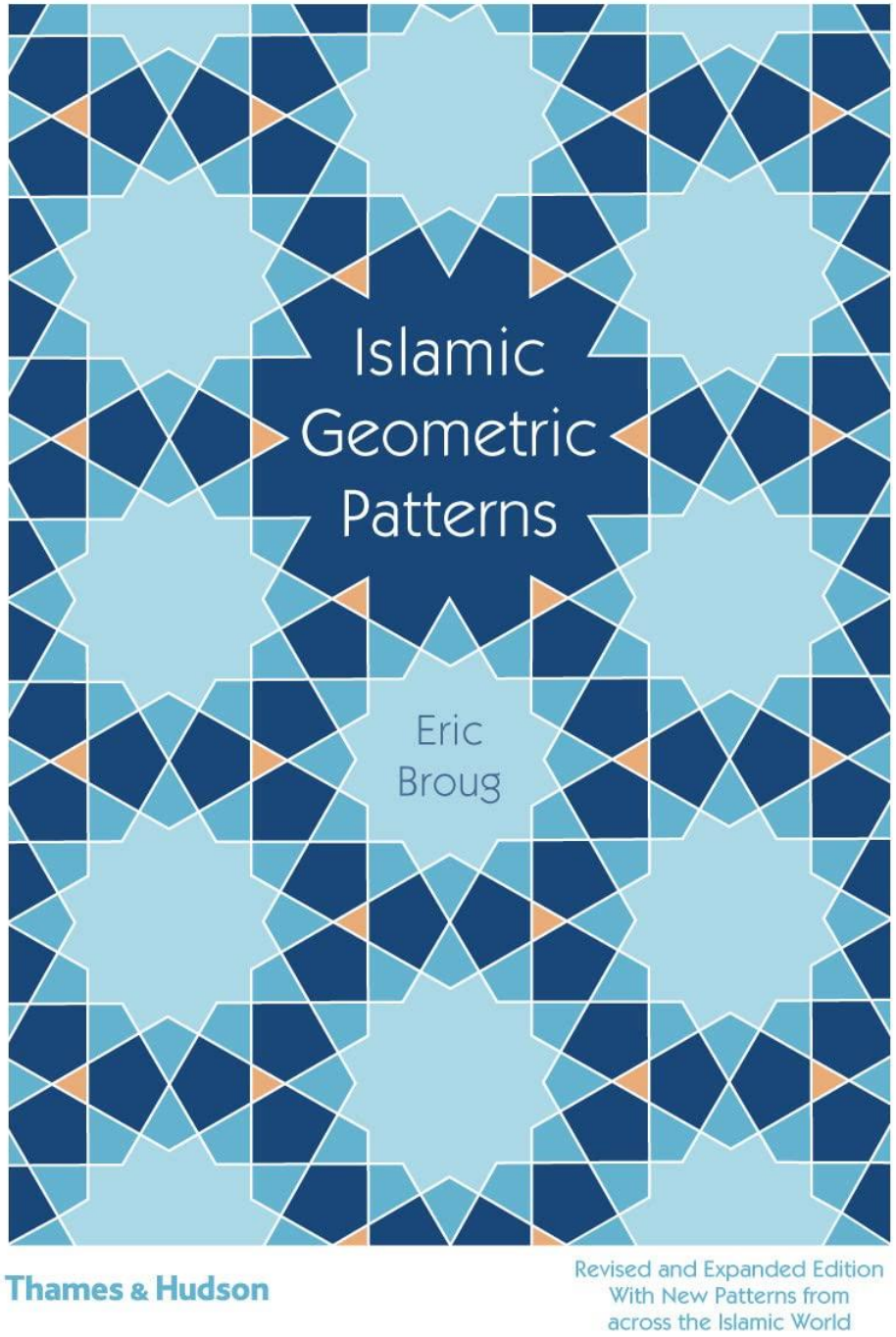
Selcuk ARTUT

Code - Tesselation

```
let radius = 20;
let motives = []; // Declare array
let nRow;
let nCol;

function setup() {
 createCanvas(800, 800);
 angleMode(DEGREES);
 stroke(0);
 nRow = 10;
 nCol = 10;
 for (let i = 0; i < nRow * nCol; i++) {
  motives.push(new Motif(radius));
 }
 noLoop();
}

function draw() {
 background(255);
 for (let r = 0; r < nRow; r++) {
  for (let c = 0; c < nCol; c++) {
   push();
   translate(6 * radius * cos(30) * c,  5 * radius * r );
   motives[r+c*nRow].display();
   pop();
  }
 }
}
```



Want more?


Islamic
Geometric
Patterns
Eric
Broug
Thames & Hudson
Revised and Expanded Edition
With New Patterns from
across the Islamic World

Thanks.
Be in touch!

Facebook/selcuk.artut
Instagram: selcukartut
Web: www.selcukartut.com
Email: selcukartut@gmail.com