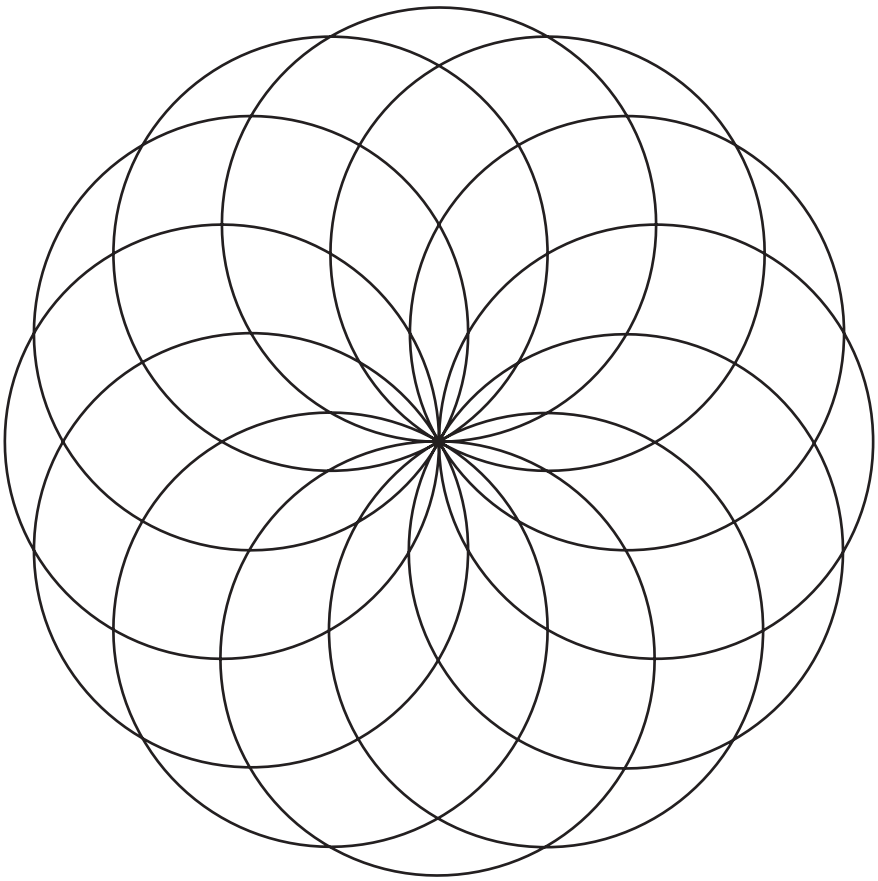
ATÖLYE

Yaratıcı Kodlama ile Geometrik Desenler

Doç. Dr. Selçuk ARTUT
Sabancı Üniversitesi, Istanbul TR
Görsel Sanatlar ve Görsel İletişim Tasarımı

Basit bir geometrik şekil üretmek

Metod 1: Cetvel ve Pergel yardımı ile el çizimi

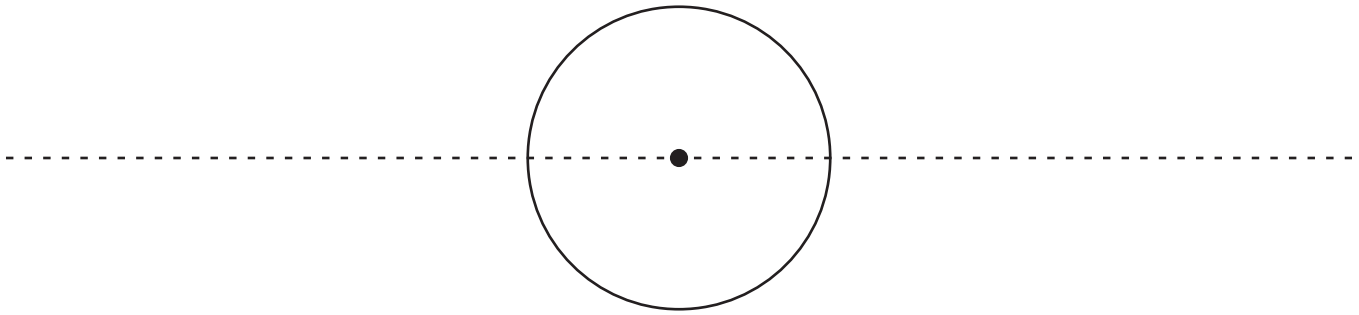


Bir daireyi 6 ve 12 eşit parçaya bölmek

Aşama 1 : Cetvel yardımı ile sayfanın ortasına düz bir çizgi çizin

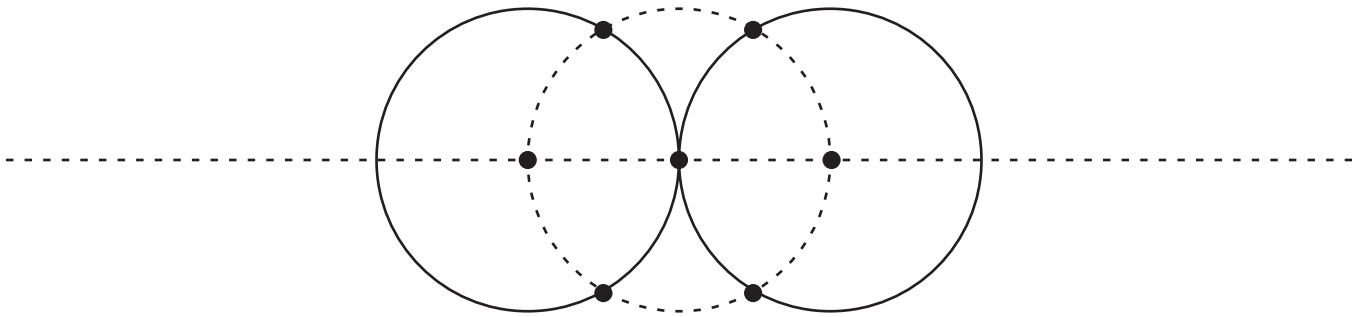
Bir daireyi 6 ve 12 eşit parçaya bölmek

Aşama 2 : Pergel yardımı ile doğrunun ortasına bir daire çizin. Pergelin çapını sakın bozmayın, ileride gerekecek.

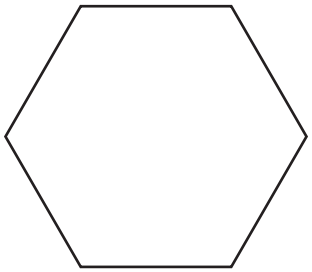


Bir daireyi 6 ve 12 eşit parçaya bölmek

Aşama 3 : İki daire daha çizin. Bu dairelerin merkezleri birinci aşamadaki daireye teğet olmalı

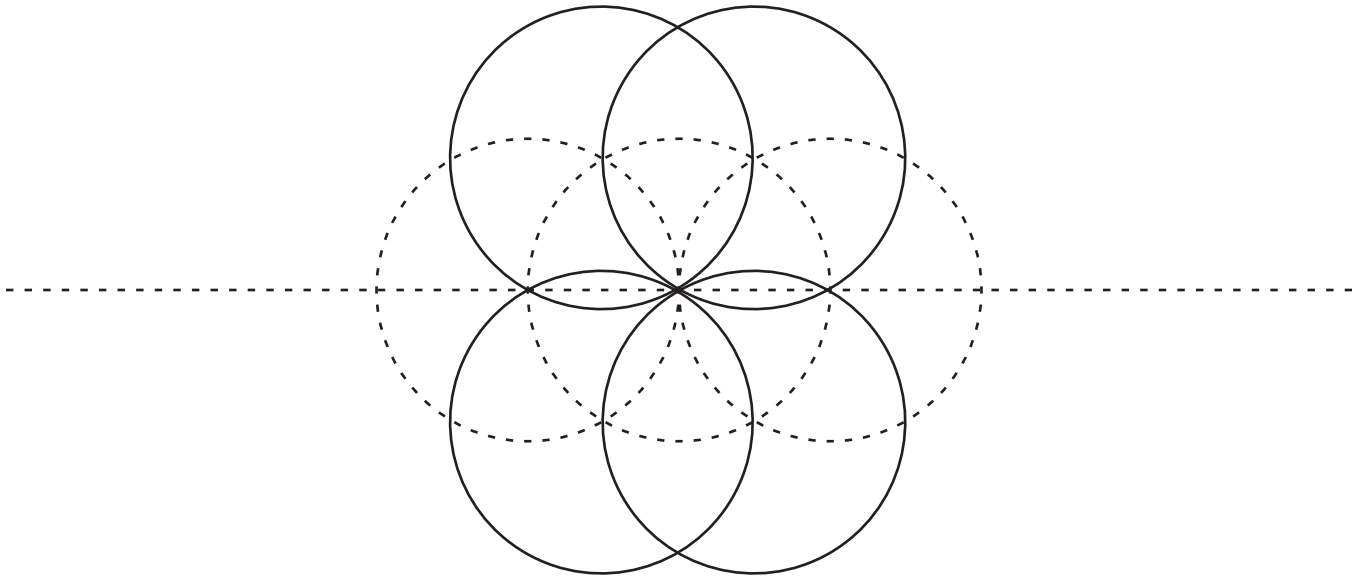


bir daireyi 6 eşit parçaya böldük



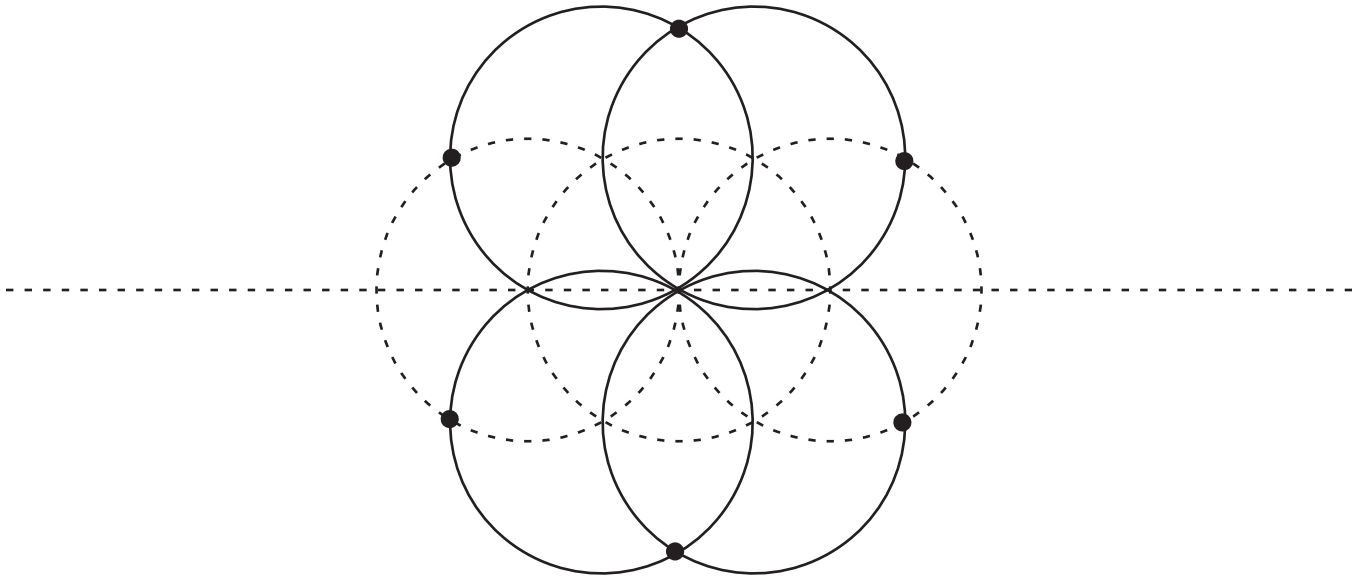
Bir daireyi 6 ve 12 eşit parçaya bölmek

Aşama 4 : Kesişim noktalarına 6 daire çizin



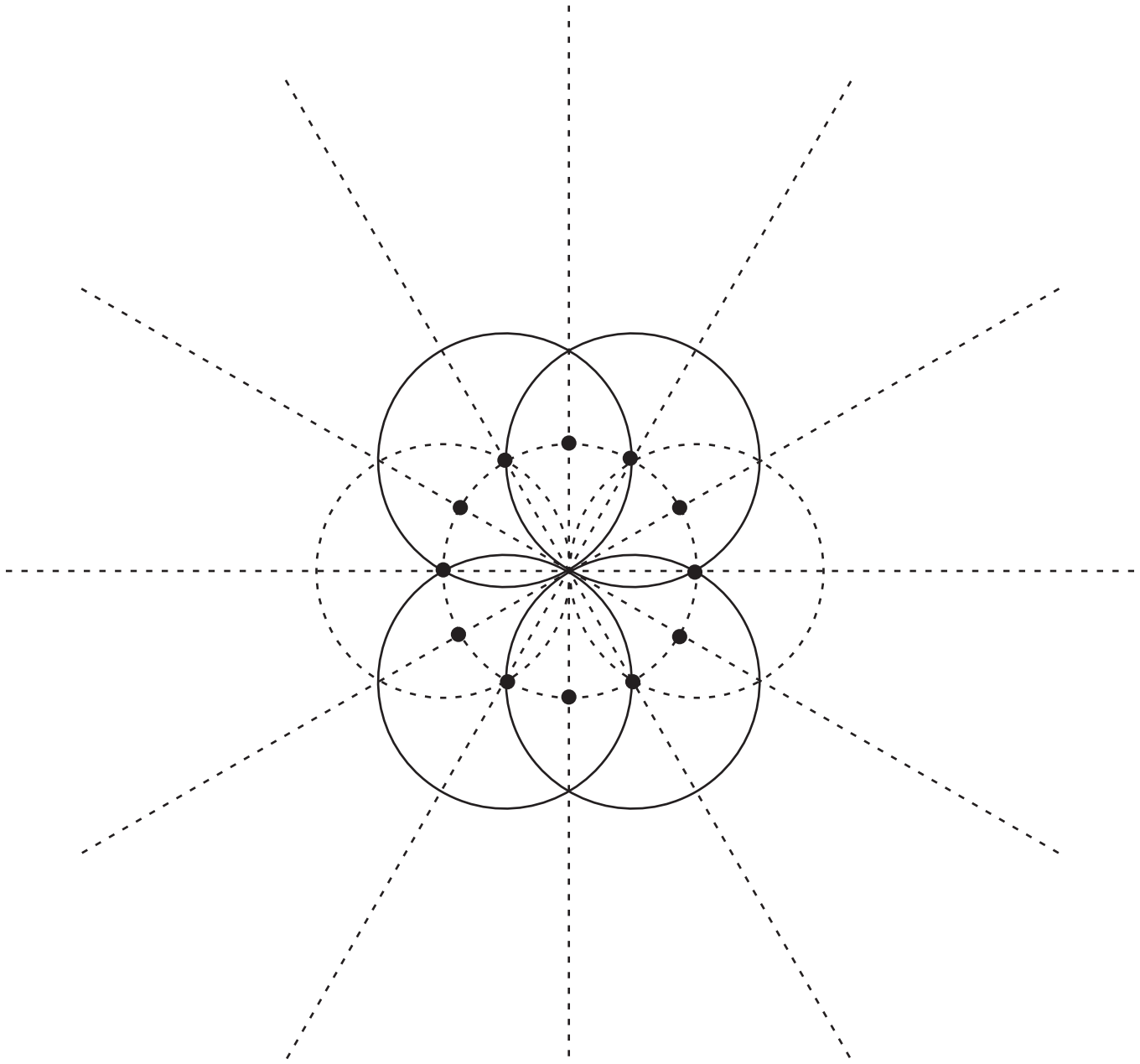
Bir daireyi 6 ve 12 eşit parçaya bölmek

Aşama 5 : Kesişim noktalarından geçen doğrular çizins

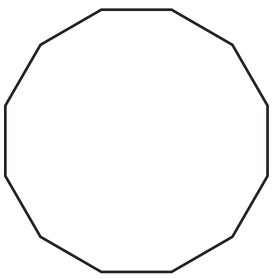


Bir daireyi 6 ve 12 eşit parçaya bölmek

Aşama 6 : Elde ettiğiniz 12 kesişim noktasına tükenmez kalem ile daire çizin

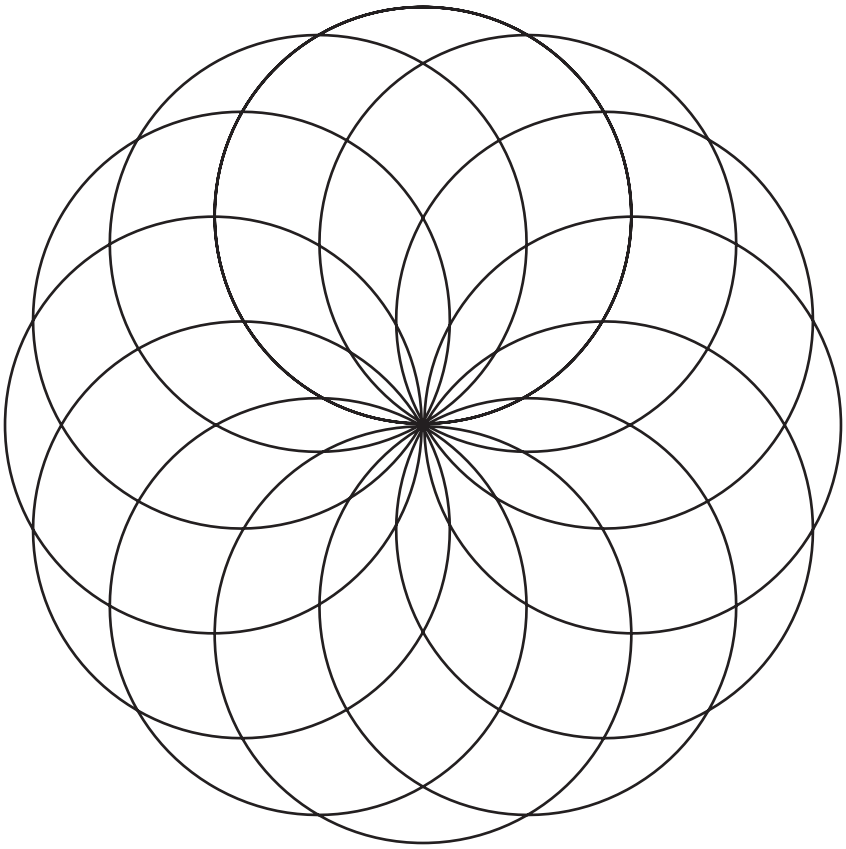


Şimdi bir daireyi 12 eşit parçaya bölmüş olduk



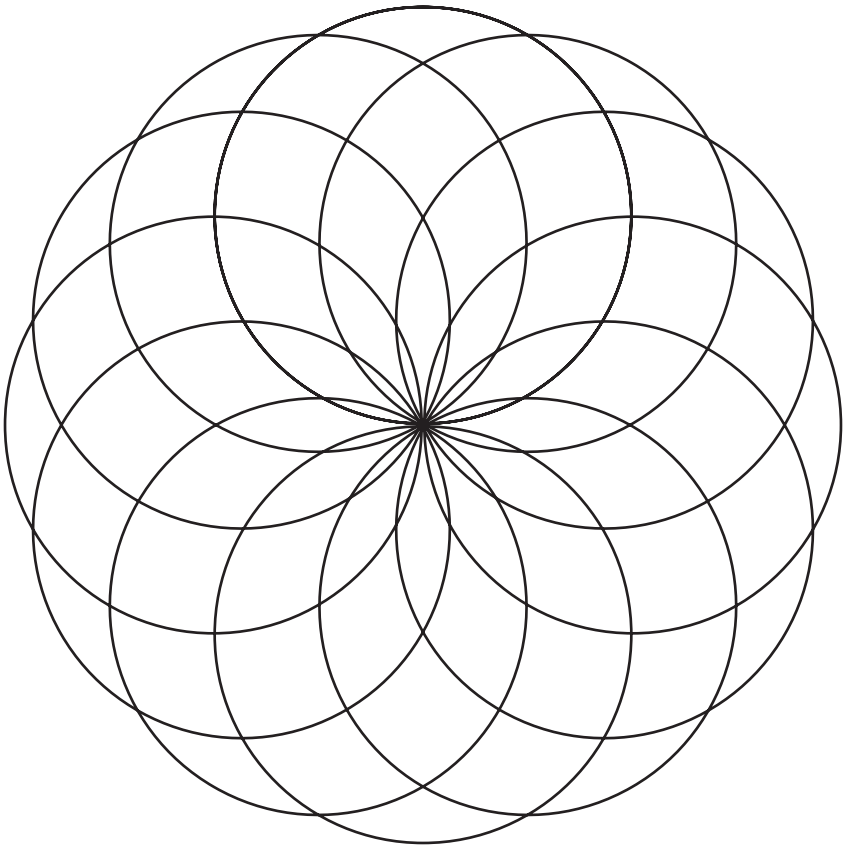
Bir daireyi 6 ve 12 eşit parçaya bölmek

Aşama 7 : Tüm kurşun kalemle yaptığınız çizgileri silebilirsiniz.



Basit bir geometrik şekil üretmek

Metod 2: Yaratıcı Kodlama Kullanarak

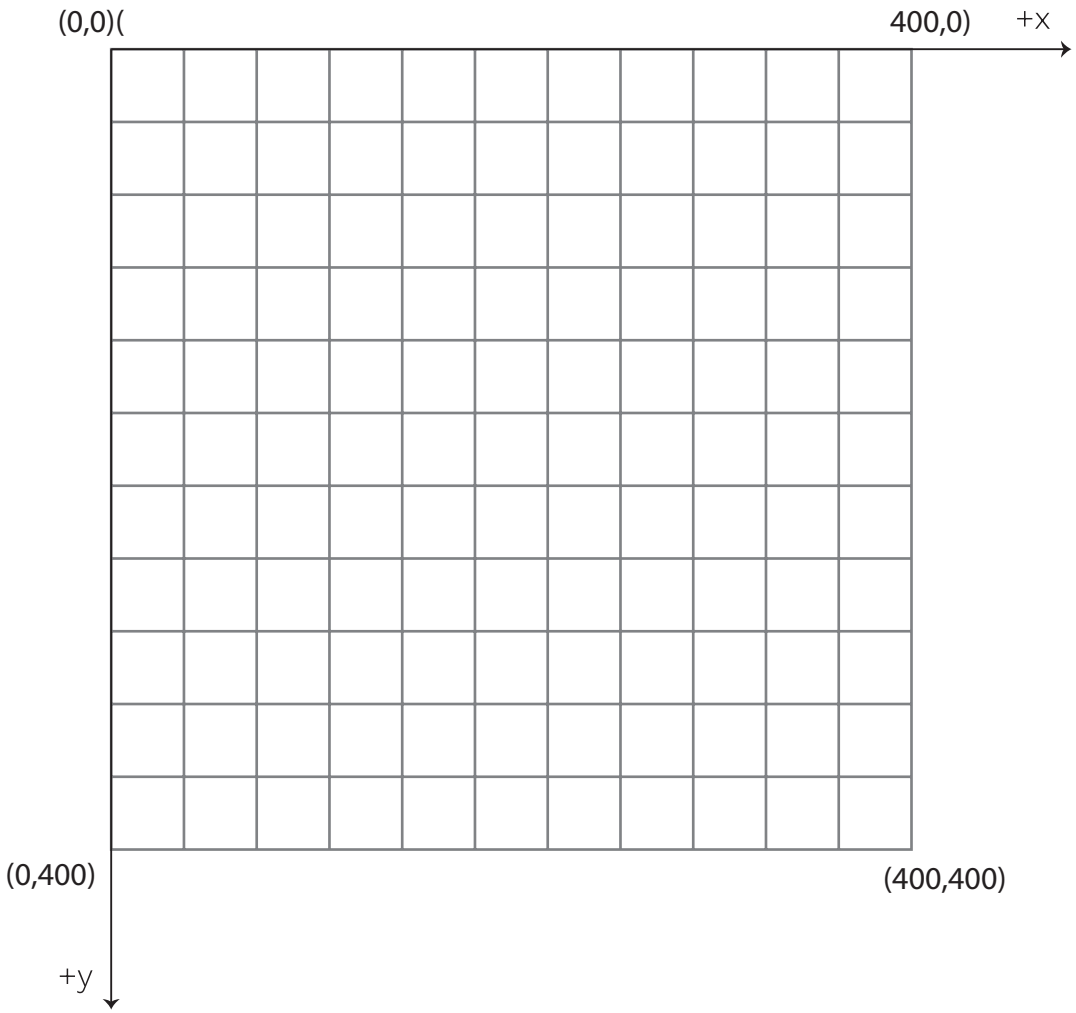


Yaratıcı Kodlama kullanarak basit bir şekil oluşturmak

Aşama 1 : Önce bir kanvas hazırlayalım.

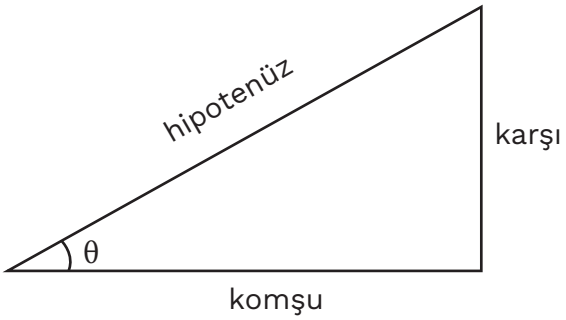
Canvas Size Width: 400px,Height: 400px
Background is white

```
function setup() {
        createCanvas(400, 400);
}
function draw() {
        background(255);
}
```



Temel Trigonometri

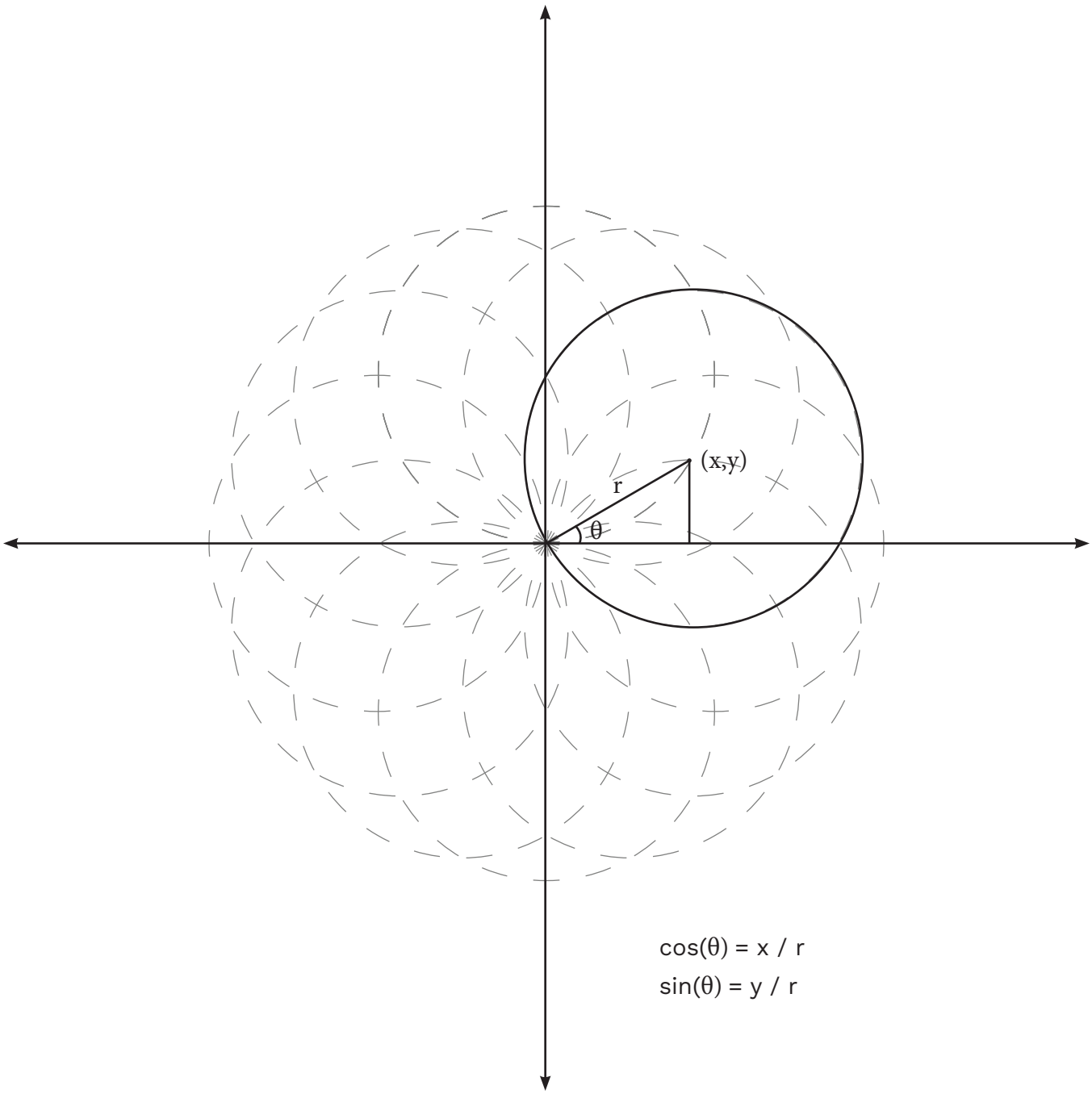
Üç temel trigonometri fonksiyonunu hatırlayalım: sinüs, cosinüs ve tanjant.



sin(θ) = karşı / hipotenüs

cos(θ) = komşu / hipotenüs

tan(θ) = karşı / komşu



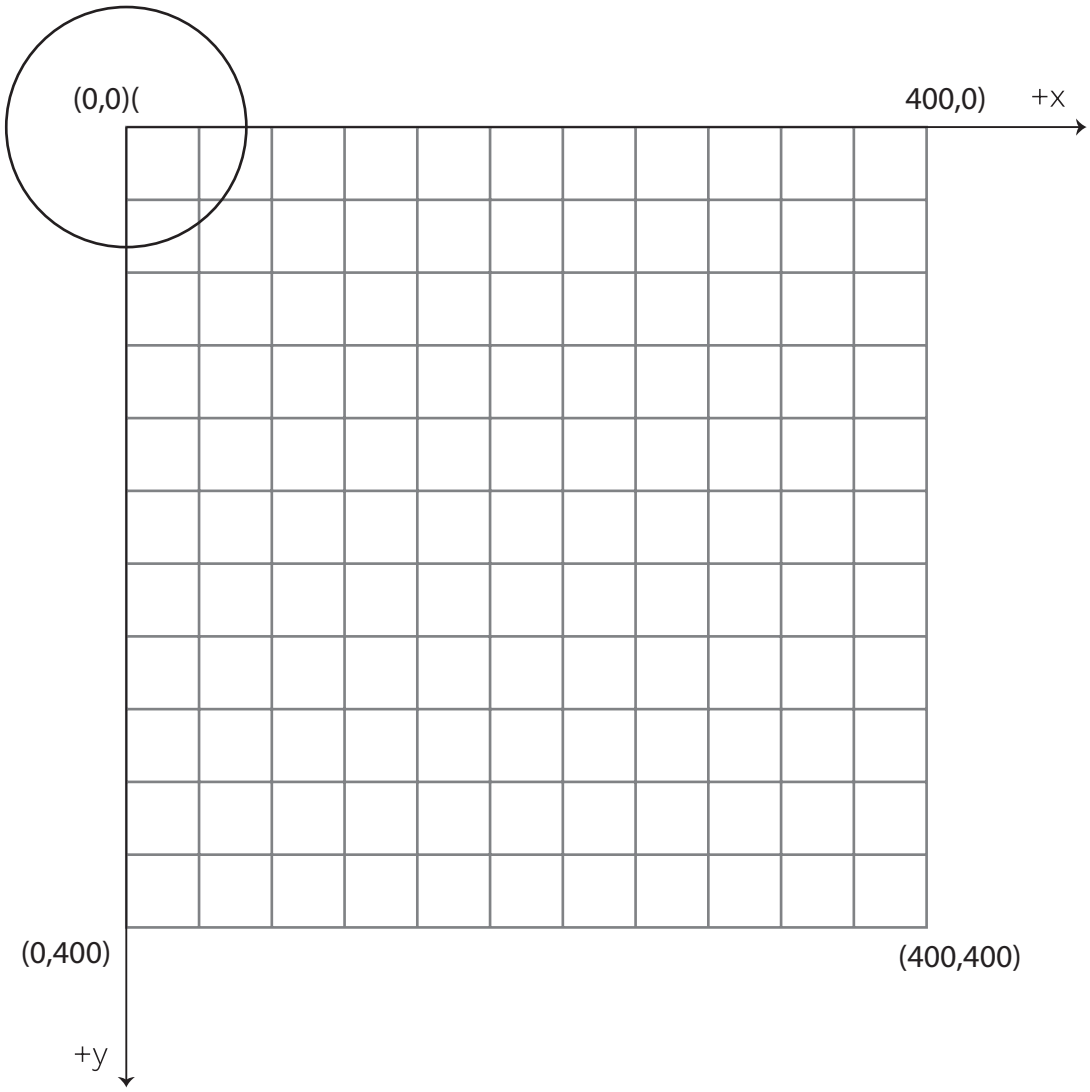
$\cos(\theta) = x / r$

$\sin(\theta) = y / r$

Yaratıcı Kodlama kullanarak basit bir şekil oluşturmak

Aşama 2 : İçi boş bir daireyi oluşturalım. Dairenin merkezi (0,0) noktasında olacaktır.

```
function setup() {
        createCanvas(400, 400);
        noFill();
}
function draw() {
        background(255);
        circle(0,0,120);
}
```

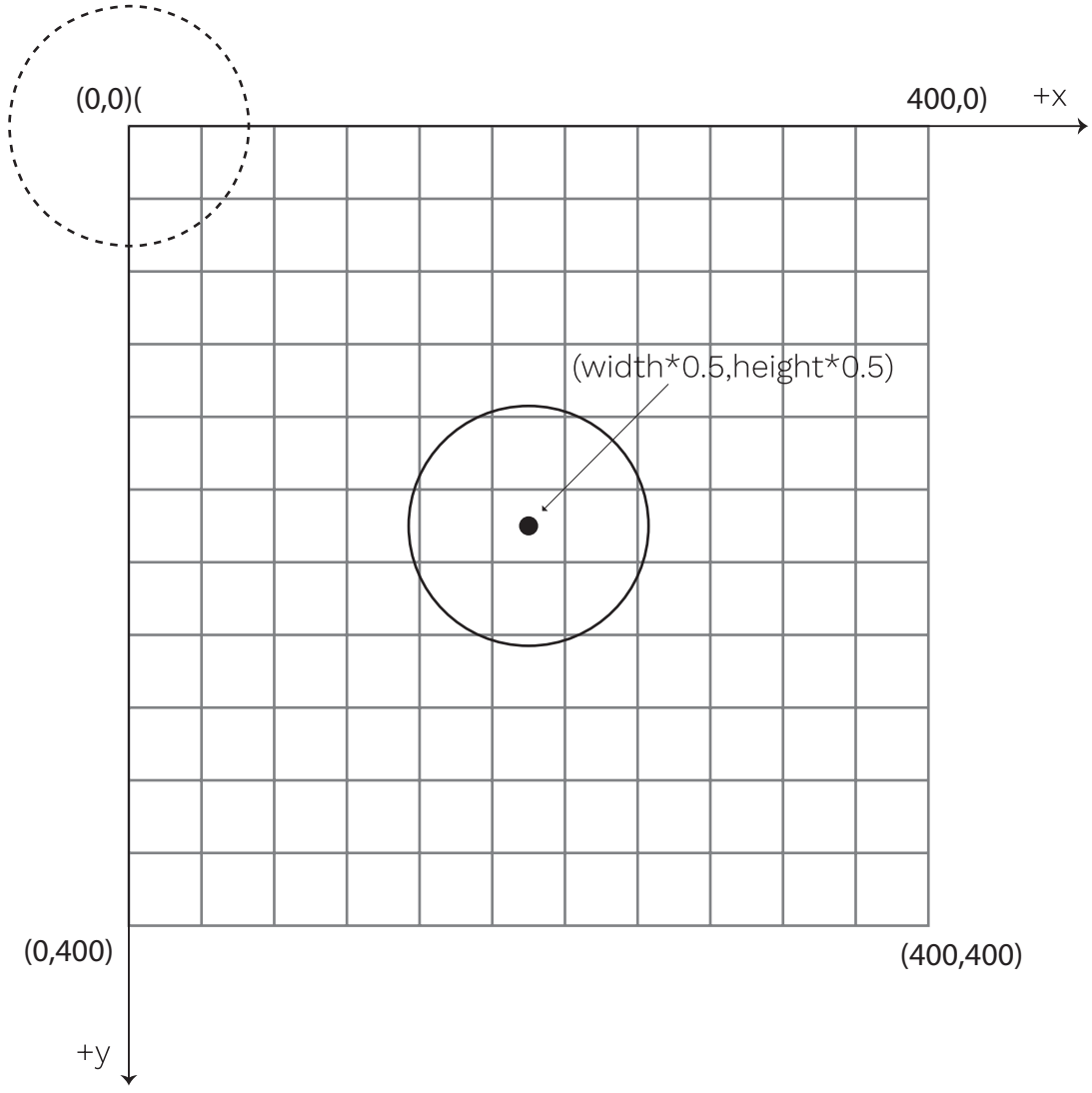


Yaratıcı Kodlama kullanarak basit bir şekil oluşturmak

Aşama 3 : Daireyi kanvasın ortasına almak için tranformasyon fonksiyonlarını kullanalım.

Kanvasın genişliğini ve yüksekliğini otomatik olarak width ve height fonksiyonları ile elde edebiliriz.

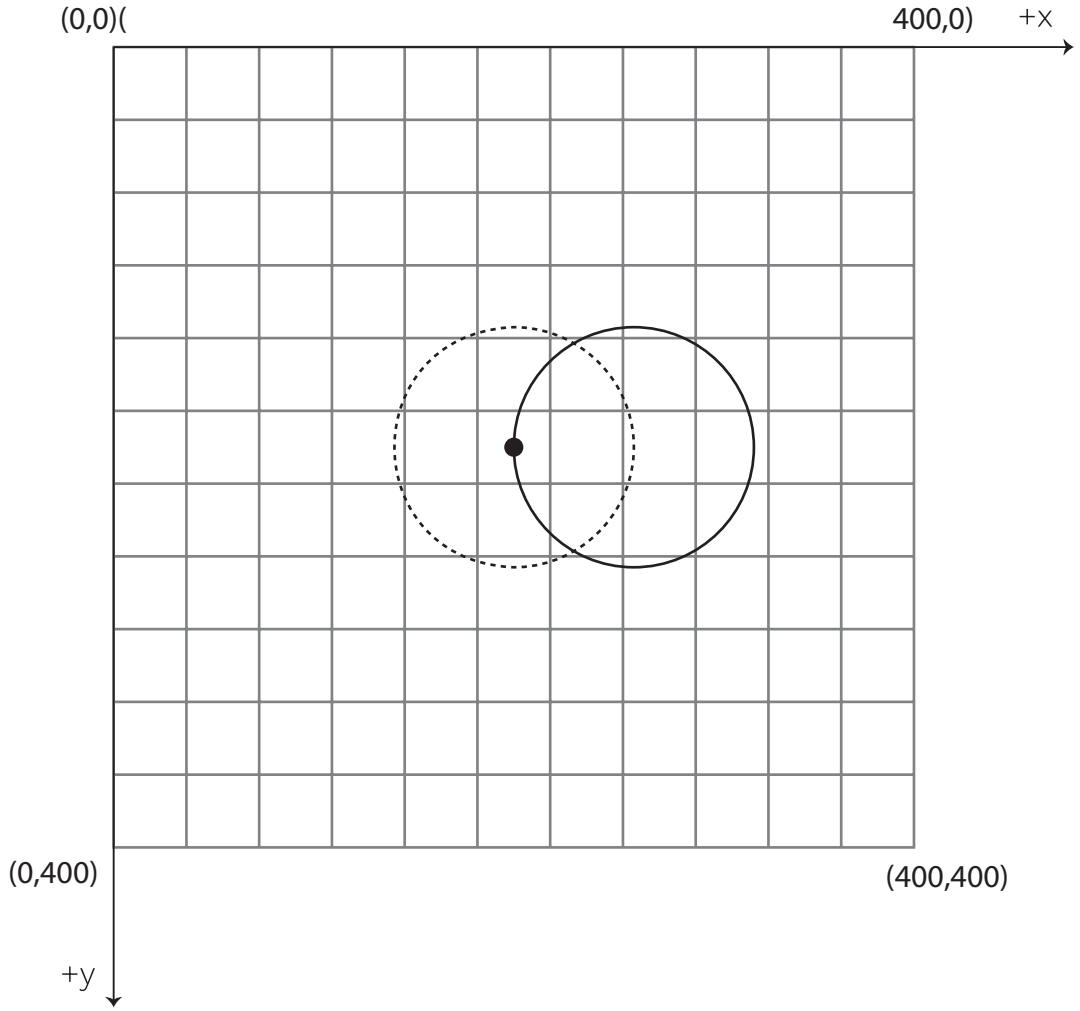
```
function setup() {
        createCanvas(400, 400);
        noFill();
}
function draw() {
        background(255);
        push();
                translate(width*0.5,height*0.5);
                circle(0,0,120);
        pop();
}
```



Aşama 4 : Dairemizi yarıçapı kadar sağa kaydıralım. Unutmayın bu örnekte çap 120 piksel.

Daha sonra bu daireden 12 tane kopya oluştururken her birini kanvasın merkezinin etrafında 30 derece döndüreceğiz.

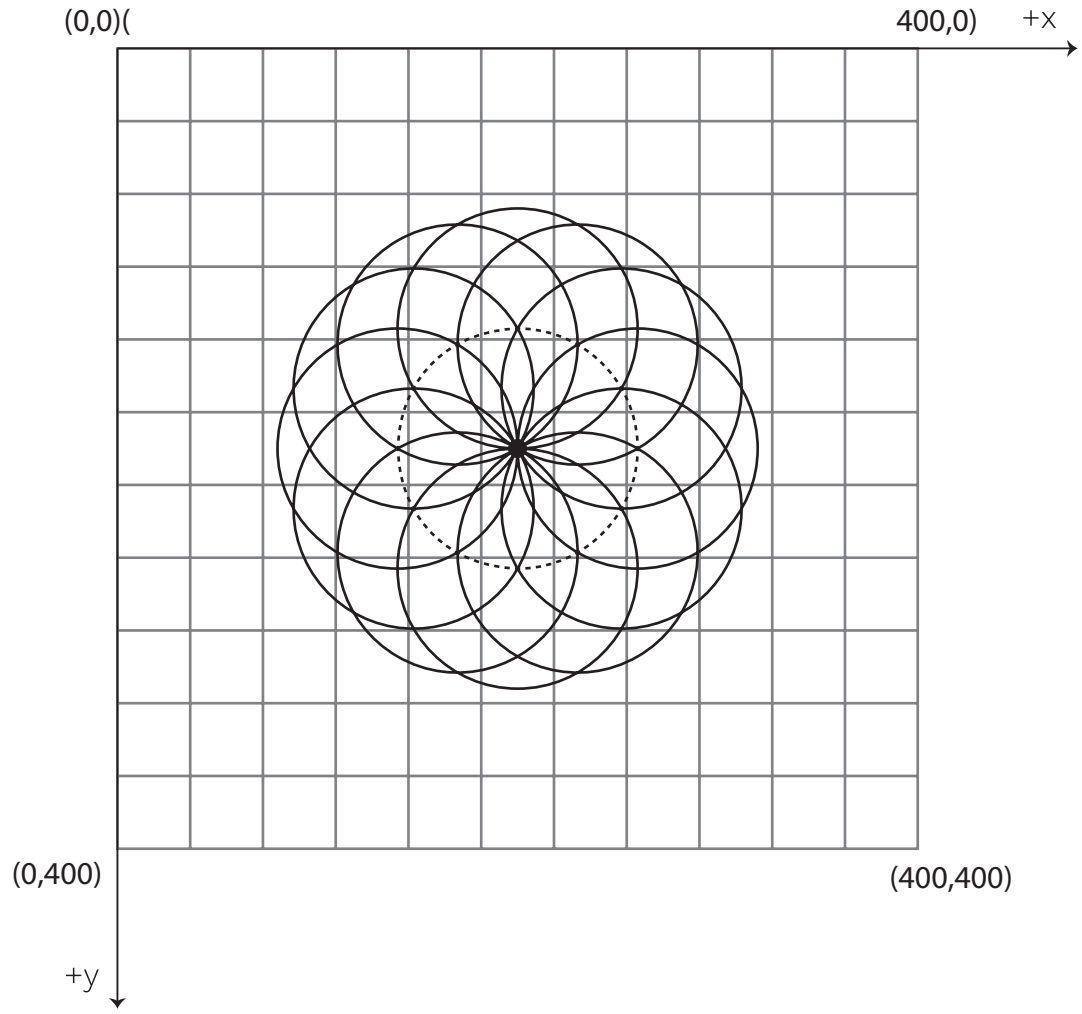
```
function setup() {
        createCanvas(400, 400);
        noFill();
}
function draw() {
        background(255);
        push();
                translate(width*0.5,height*0.5);
                push();
                        translate(60,0);
                        circle(0,0,120);
                pop();
        pop();
}
```



Yaratıcı Kodlama kullanarak basit bir şekil oluşturmak

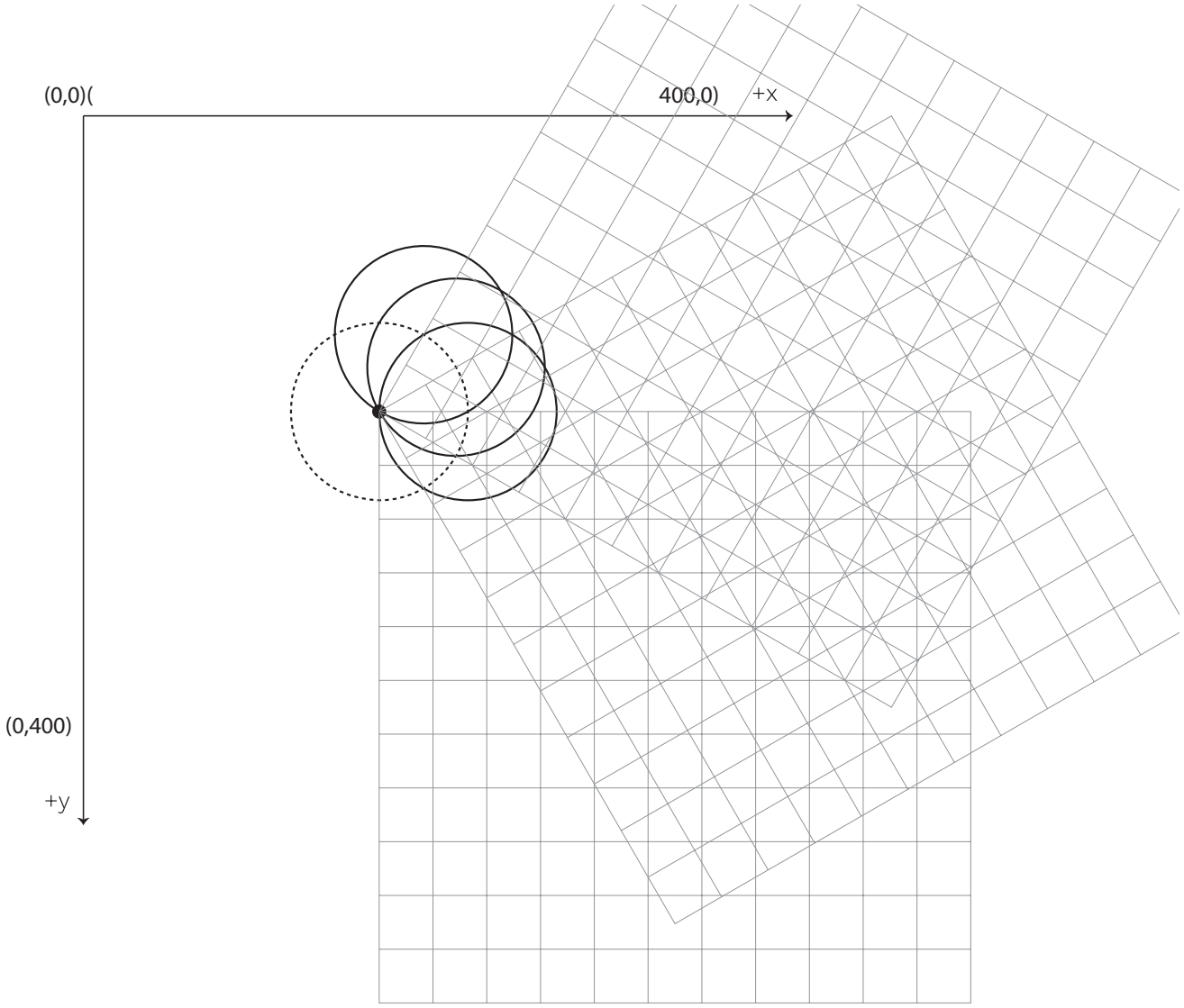
Aşama 5 : Şimdi “for” döngüsü kullanarak 12 tane daire kopyası oluşturalım. Ancak açı moduna dikkat, çünkü p5.js ortamında aksini belirtmezseniz radian birimi

```
function setup() {
        createCanvas(400, 400);
        angleMode(DEGREES);
        noFill();
}
function draw() {
        background(255);
        push();
                translate(width*0.5,height*0.5);
                for(let i = 0; i < 12; i++){
                push();
                        rotate(i*30);
                        translate(60,0);
                        circle(0,0,120);
                pop();
                }
        pop();
}
```



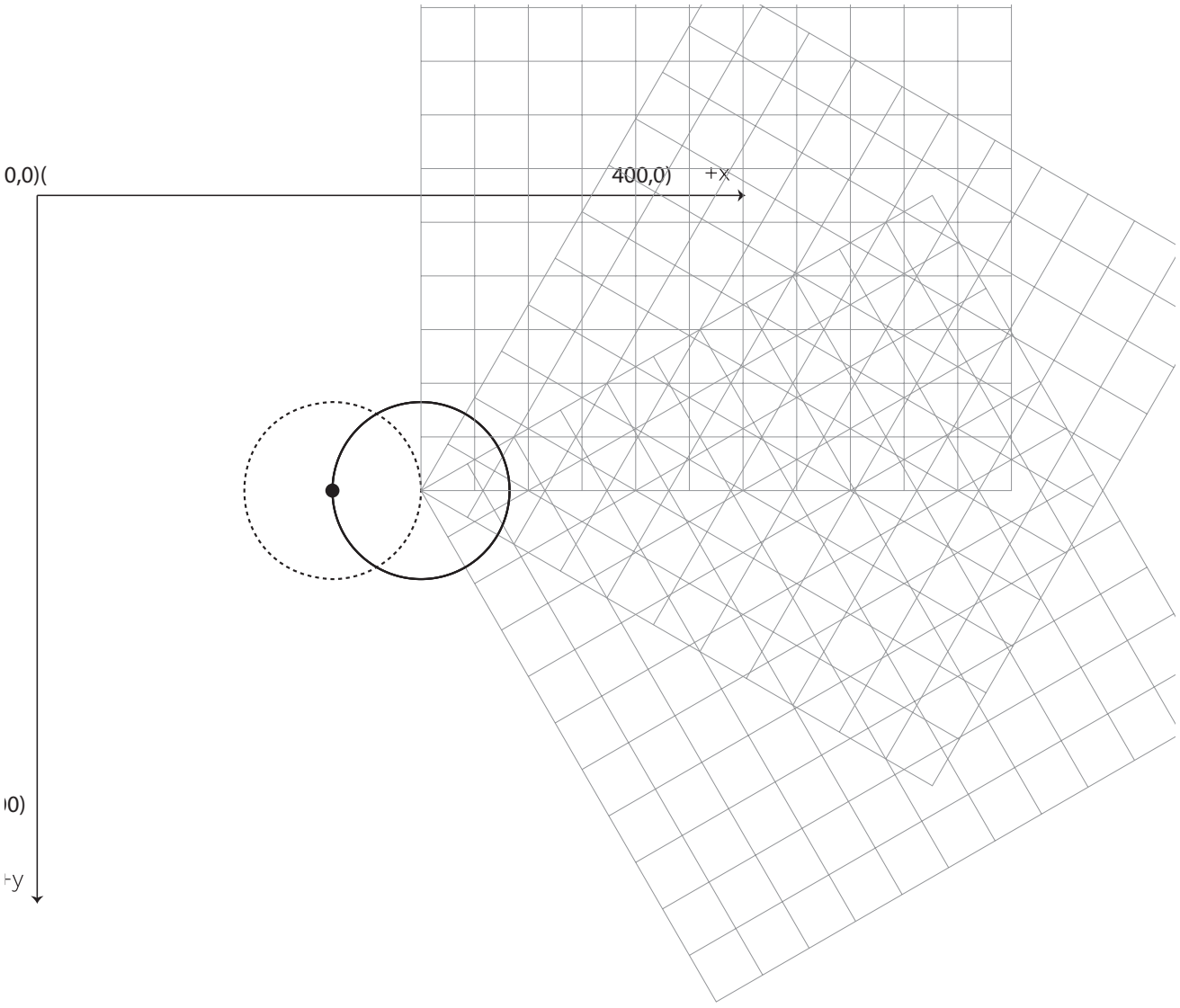
Yaratıcı Kodlama kullanarak basit bir şekil oluşturmak

Transformation sıralaması önemli!!!
Aşağıda görsel bir karmaşa oluşmaması için 3 iterasyon göreceksiniz.
Önce Rotate sonra Translate yaparsak;



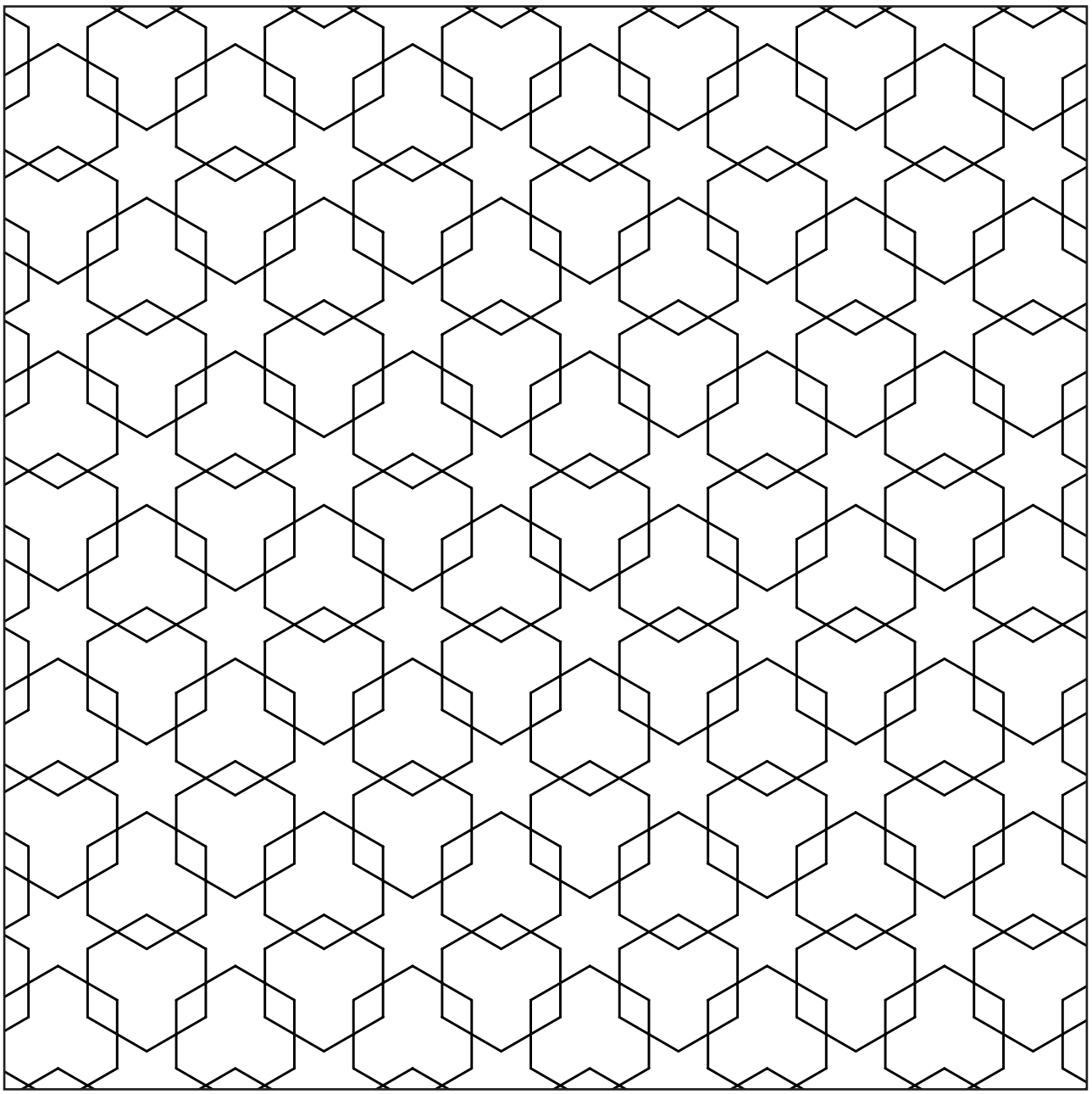
Yaratıcı Kodlama kullanarak basit bir şekil oluşturmak

Transformation sıralaması önemli!!!
Aşağıda görsel bir karmaşa oluşmaması için 3 iterasyon göreceksiniz.
Önce Translate sonra Rotate yaparsak;

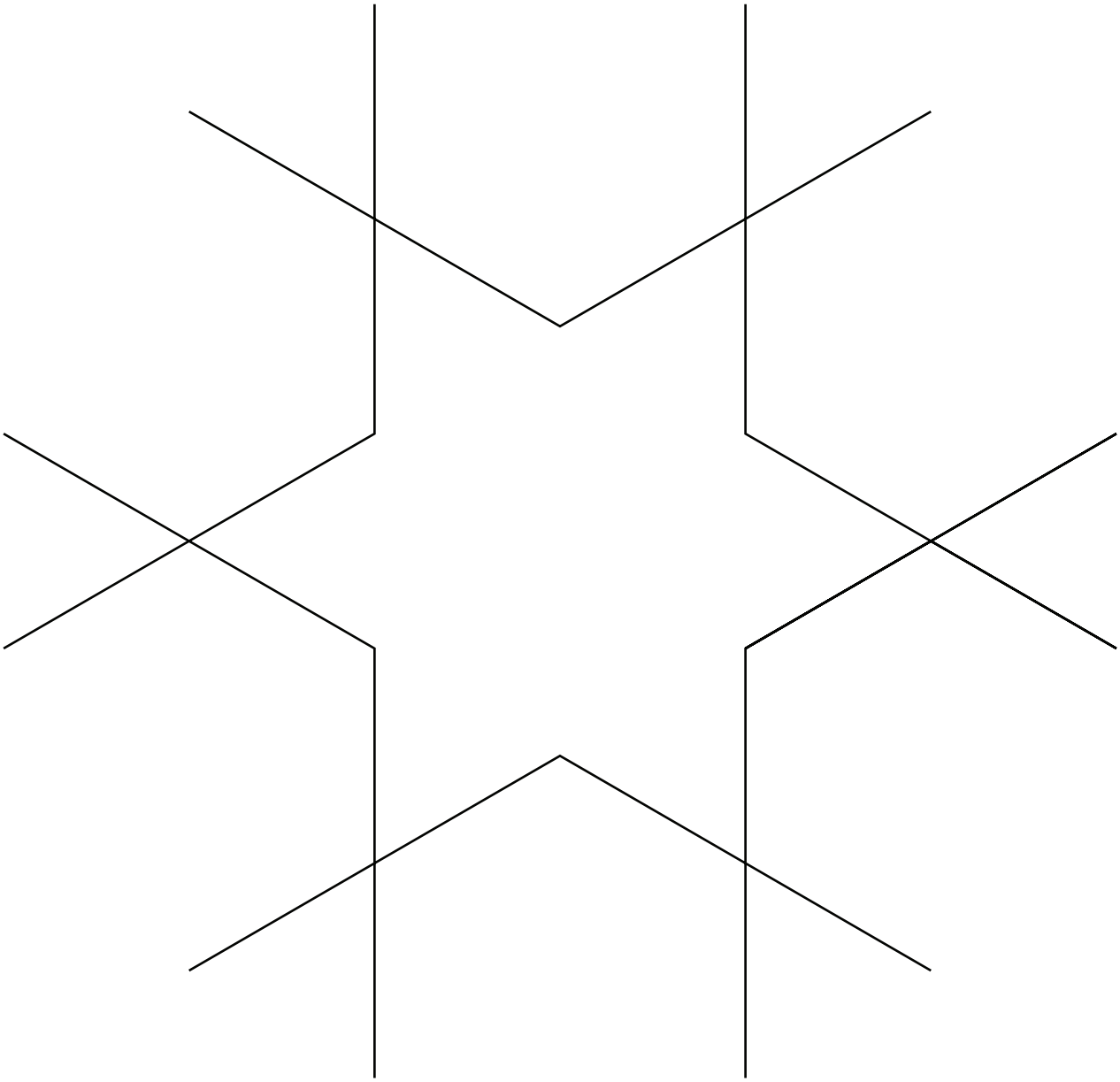


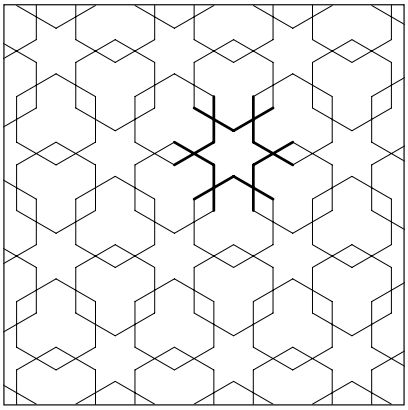
Geometrik Deseni Kodlamak
Örnek 1: Mevlana Müzesi, Konya

Aşağıdaki deseni inceleyin ve bu deseni oluşturan temel görsel bileşeni bulmaya çalışın

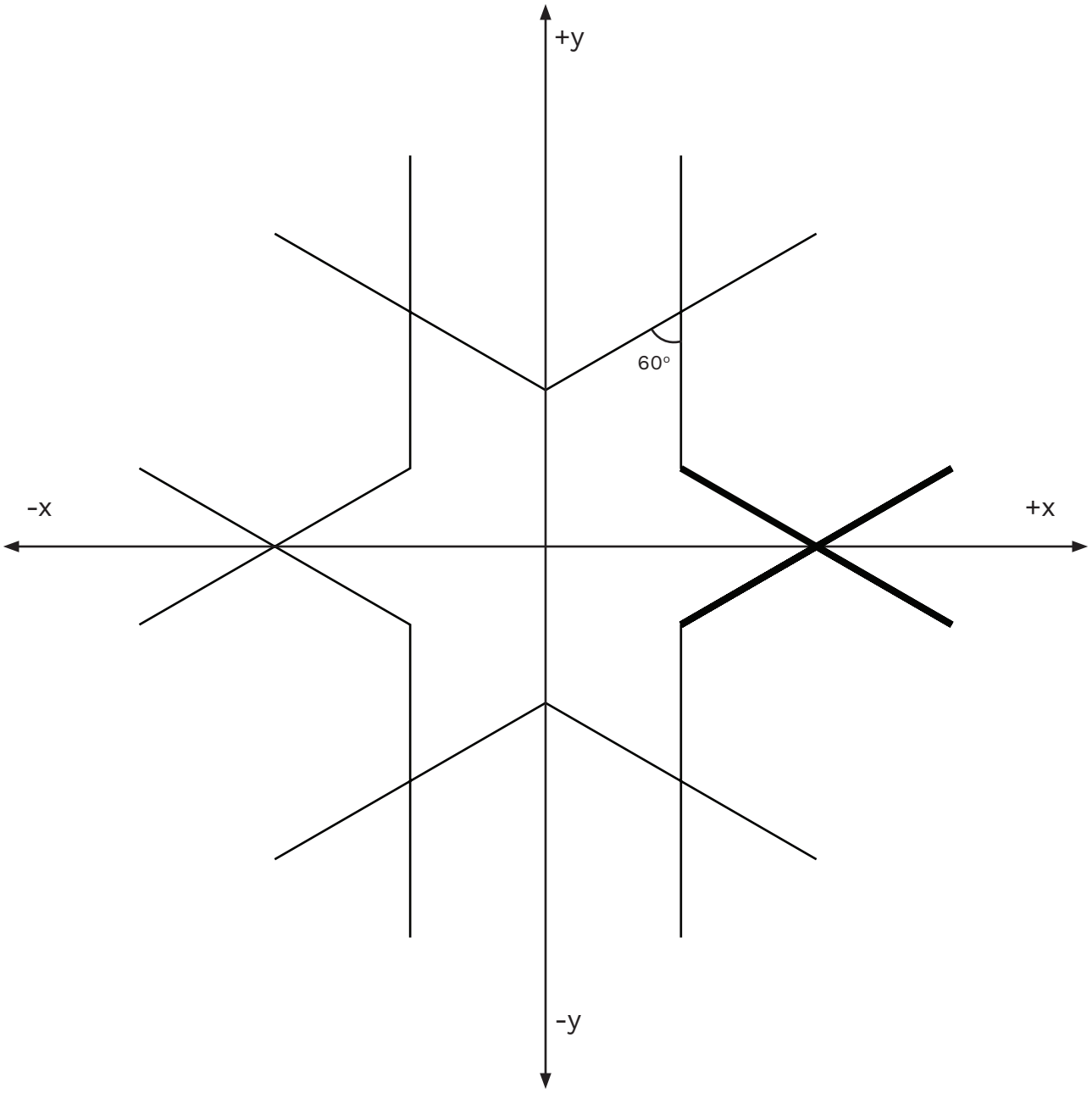


Motif





Temel Görsel Bileşini İnceleyelim



Açıları ve Vertex noktalarını tespit etmek

Aşama 1 : Vertex noktalarını bulalım

$x_0 = a \times \cos(30^\circ)$

$y_0 = a \times \sin(30^\circ)$

$x_1 = -x_0$

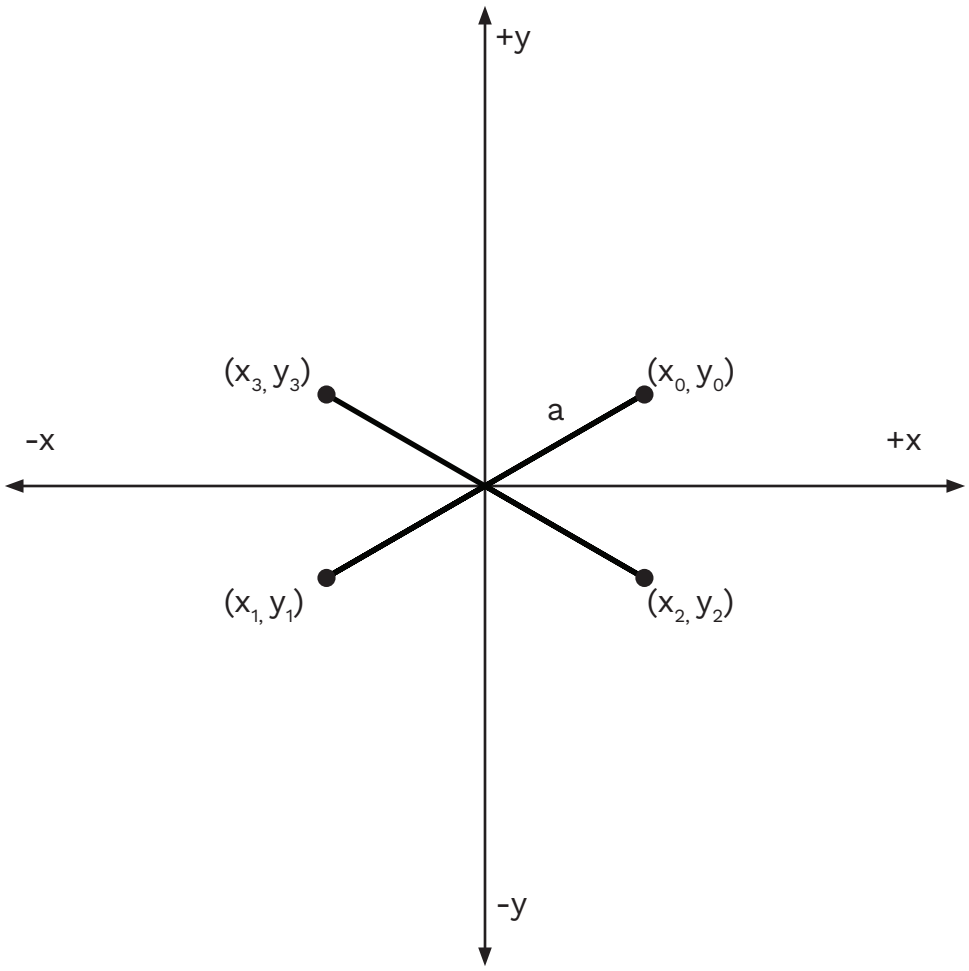
$y_1 = -y_0$

$x_2 = x_0$

$y_2 = y_1$

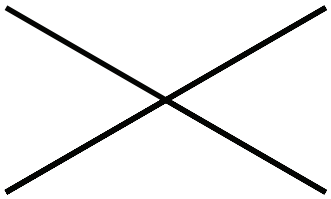
$x_3 = x_1$

$y_3 = y_0$



Motifi Oluşturmak

Aşama 2 : Temel Görsel Bileşeni çizmeye çalışalım. İki tane kesişen doğrudan oluşuyor.

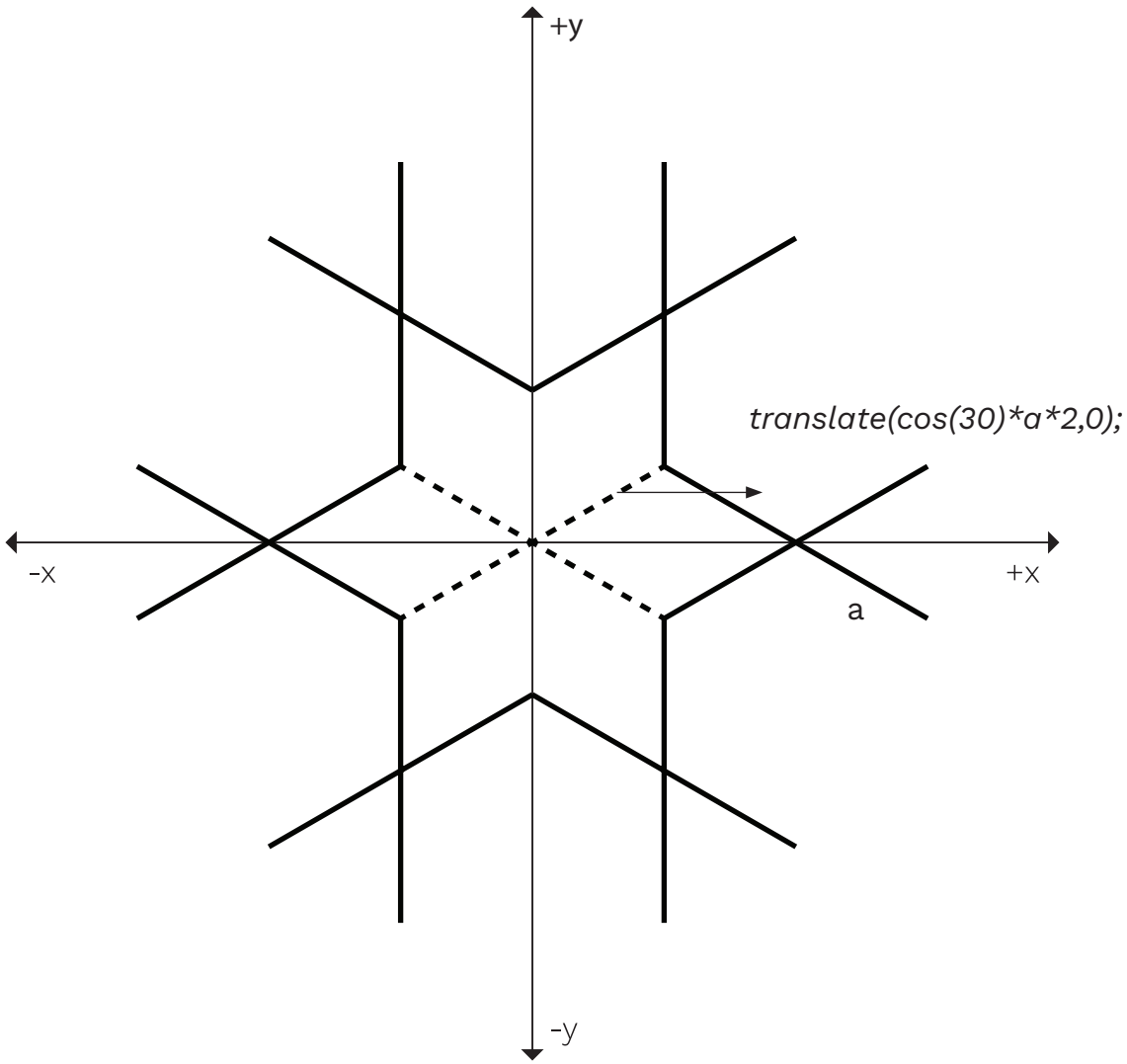


```
//scale factor
let a = 60;

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

function draw() {
   let x0,y0,x1,y1,x2,y2,x3,y3;
   push();
      translate(width*0.5, height*0.5);
      //move to the right by its width size
      //line one
      beginShape();
      x0 = a * cos(30);
      y0 = a * sin(30);
      vertex(x0,y0);
      x1 = -1 * x0;
      y1 = -1 * y0;
      vertex(x1,y1);
      endShape();
      //line two
      beginShape();
      x2 = x0;
      y2 = y1;
      vertex(x2,y2);
      x3 = x1;
      y3 = y0;
      vertex(x3,y3);
      endShape();
   pop();
}
```

Motifi Oluşturmak

Aşama 3 : Motifi, Temel Görsel Bileşeni kullanarak transformasyon fonksiyonları yardımı ile oluşturalım.

Algoritma: Temel Görsel Bileşeni yarı genişliğinde sağa kaydır. Merkez etrafında altı defa döndür.

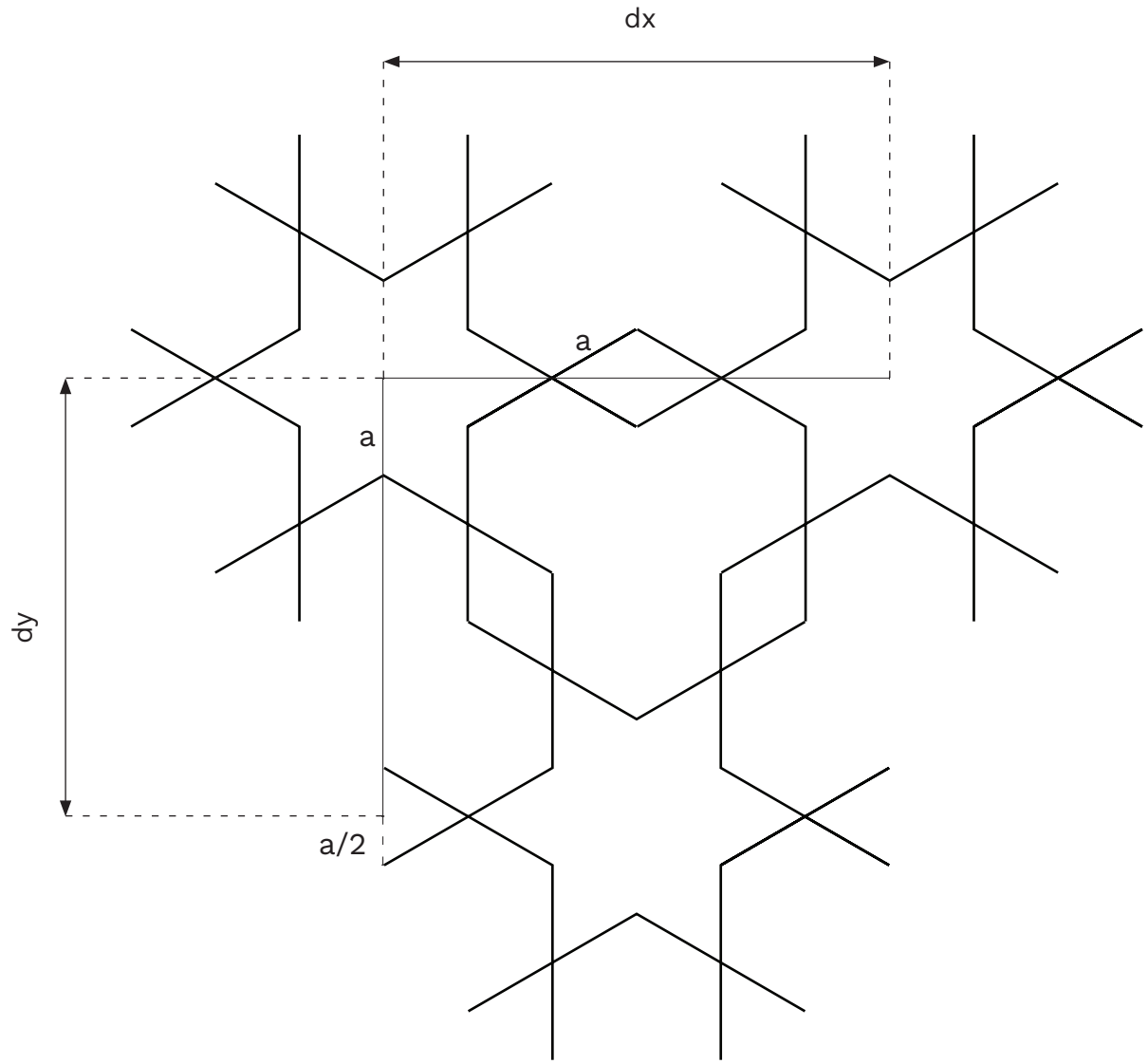


Motifi Oluşturmak

Aşama 4 : Motifi kodla oluşturmak için 6 tekrarlı bir loop döngüsü kullanacağız. Unutmayın transformasyon fonksiyonlarının sırası önemli!

```
//scale factor
let a = 60;

function setup() {
    createCanvas(400, 400);
    angleMode(DEGREES);
    noFill();
    noLoop();
}
function draw() {
    let x0,y0,x1,y1,x2,y2,x3,y3;
    push();
    translate(width*0.5, height*0.5);
        for(let i=0; i<6; i++){
            push();
                    rotate(i*60);
                    //move to the right by its width size
                    translate(cos(30)*a*2,0);
                    //line one
                    beginShape();
                    x0 = a * cos(30);
                    y0 = a * sin(30);
                    vertex(x0,y0);
                    x1 = -1 * x0;
                    y1 = -1 * y0;
                    vertex(x1,y1);
                    endShape();
                    //line two
                    beginShape();
                    x2 = x0;
                    y2 = y1;
                    vertex(x2,y2);
                    x3 = x1;
                    y3 = y0;
                    vertex(x3,y3);
                    endShape();
                pop();
            }
    pop();
}
```

Bezeme Yapısını İnceleyelim



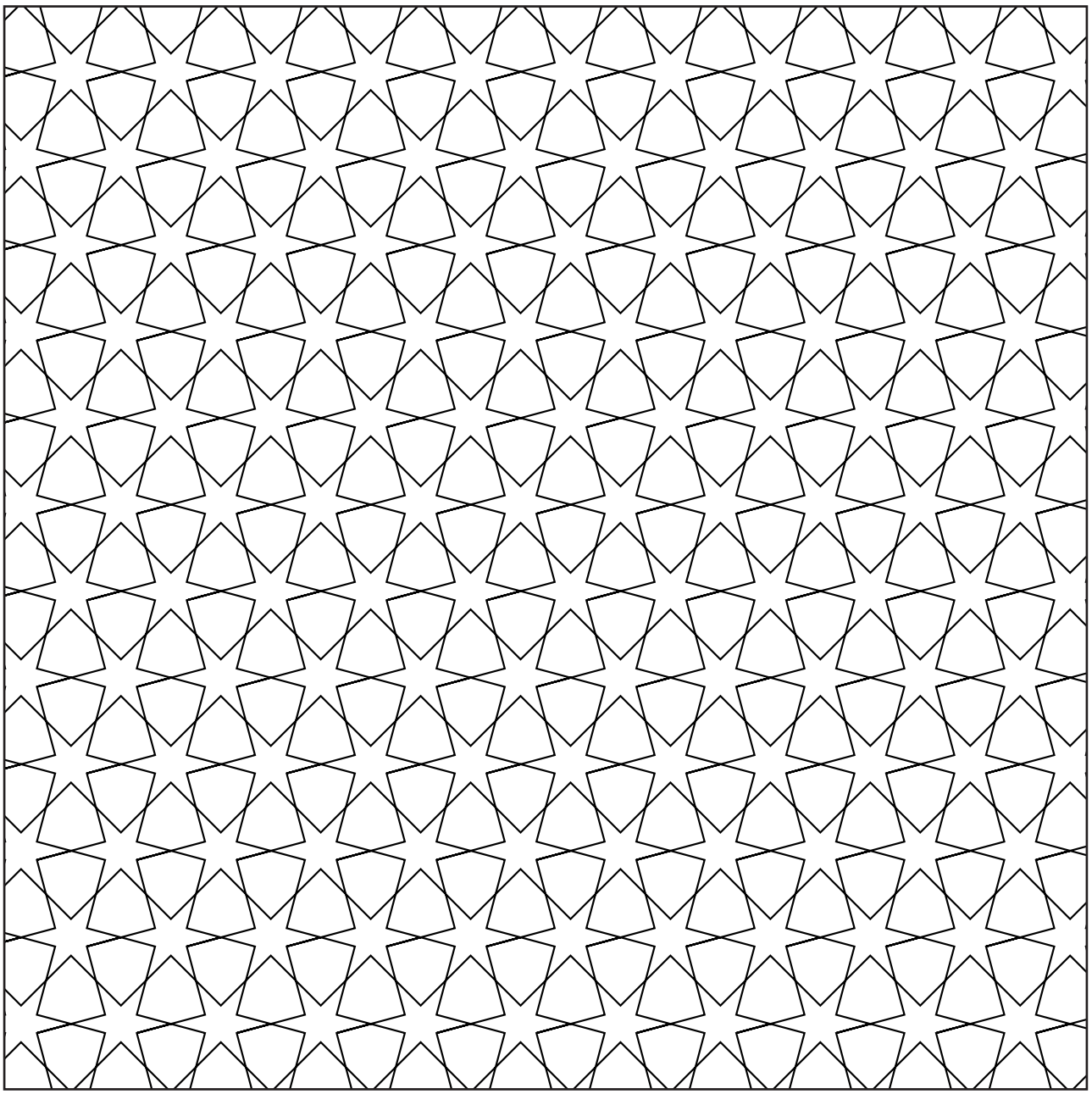
Bezeme Kodu

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

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

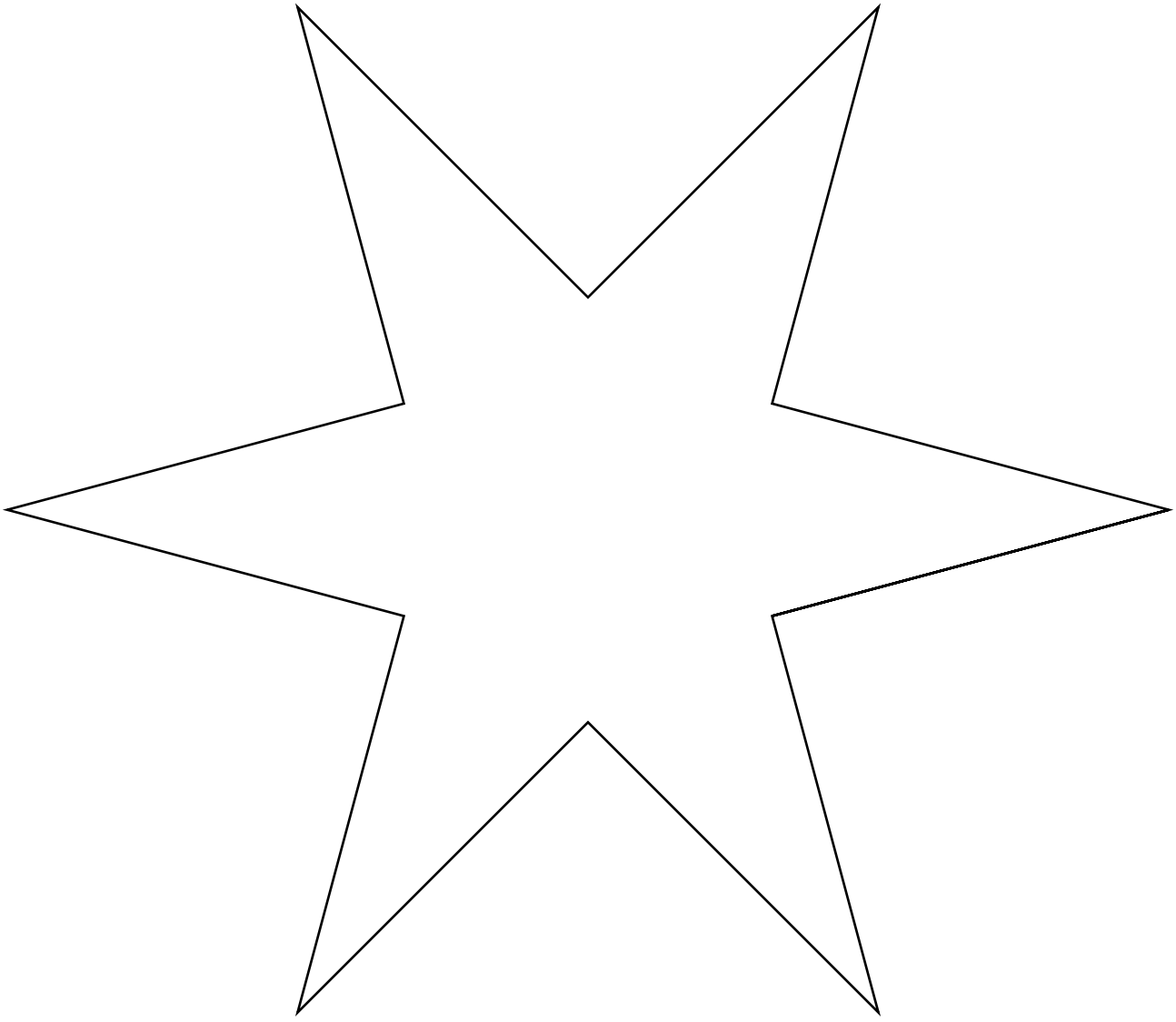
```
//scale factor
let a = 24;
let motif = new Motif(a);
let nRow;
let nCol;
let dx, dy;

function setup() {
    createCanvas(800, 800);
    angleMode(DEGREES);
    noFill();
    noLoop();

    dx = 6 * a * cos(30);
    dy = 4.5 * a;
    doff = 0.5 * dx;

    //approximate the nRow and nCol values
    nCol = ceil(width / dx);
    nRow = ceil(height / dy);
}

function draw() {
    for (let c = 0; c < nCol; c++) {
        for (let r = 0; r < nRow; r++) {
            push();
            if (r % 2 == 0) {
                //columns 0,2,4
                translate(doff, 0);
            }
            translate(c * dx, r * dy);
            motif.display();
            pop();
        }
    }
}
```

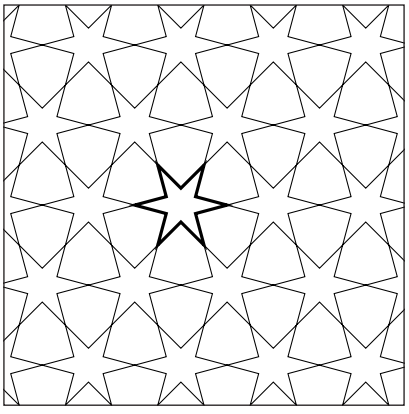
Geometrik Deseni Kodlamak
Örnek 2: Eşrefoğlu Camii, Beyşehir

Aşağıdaki deseni inceleyin ve bu deseni oluşturan temel görsel bileşeni bulmaya çalışın

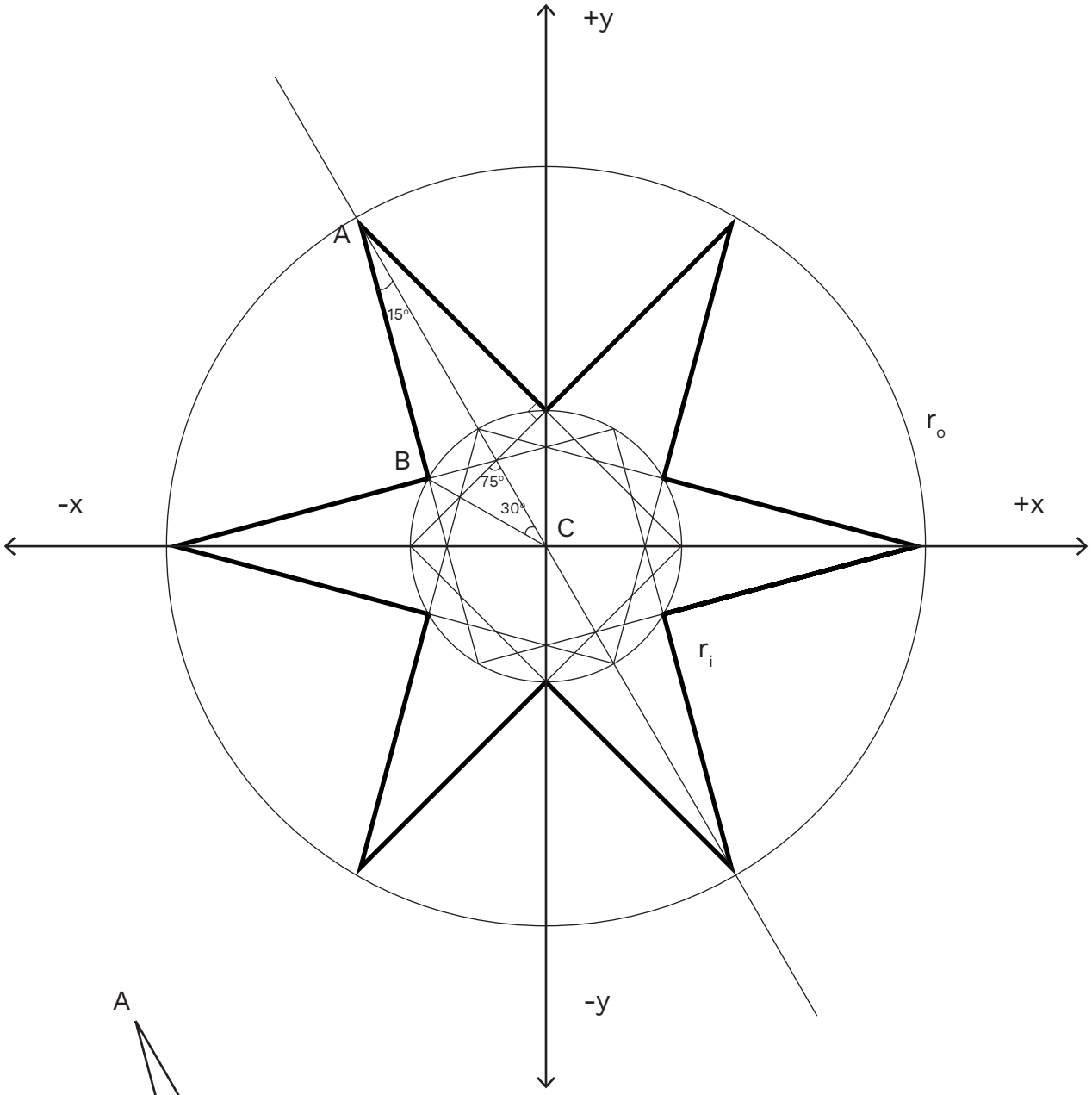


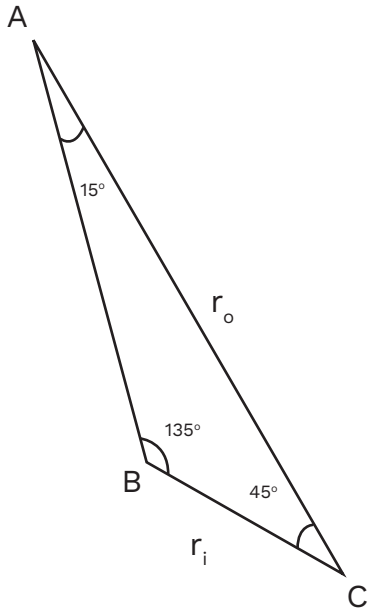
Motif





Temel Görsel Bileşini İnceleyelim

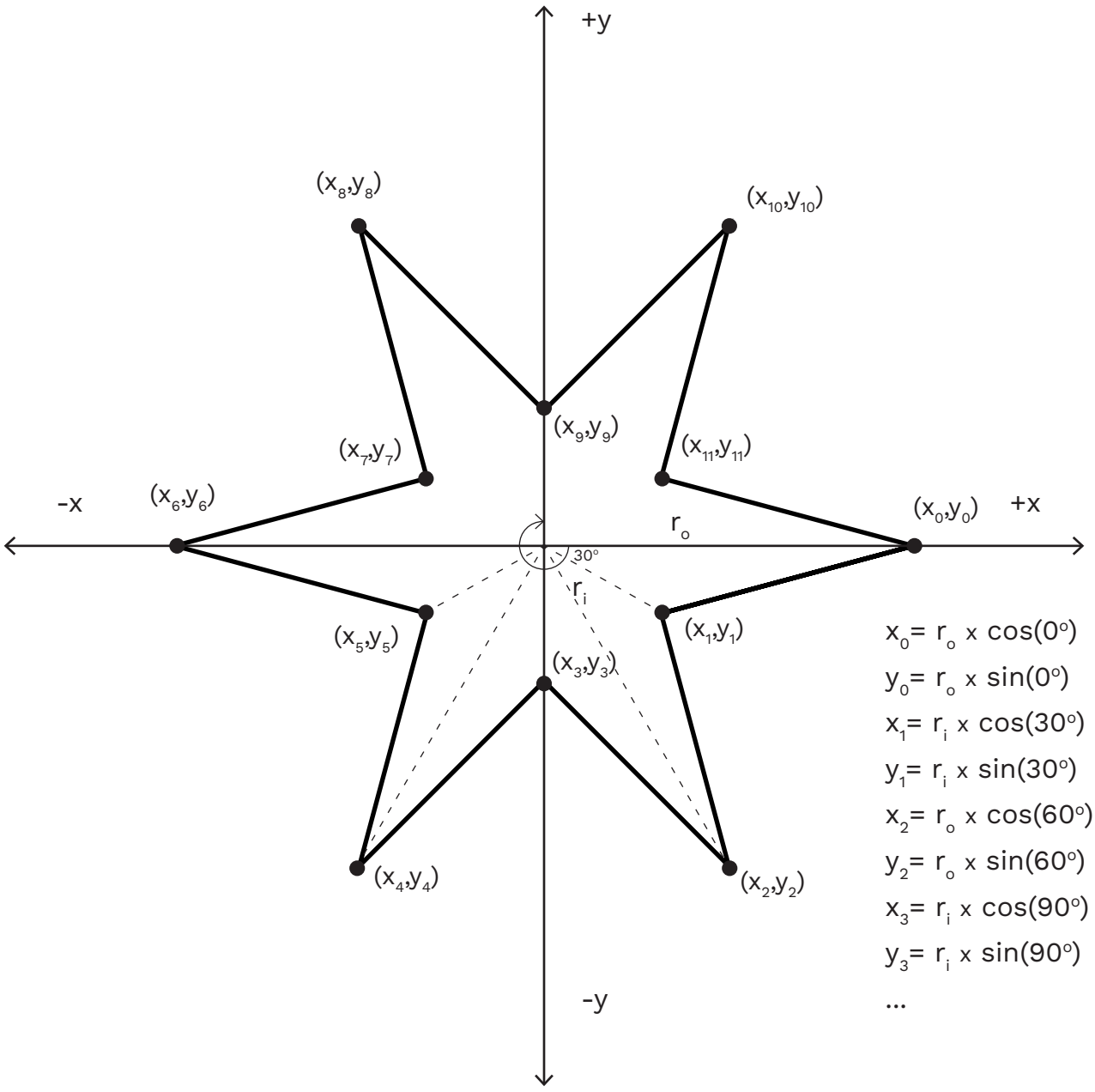




$$\frac{r_i}{\sin(15^\circ)} = \frac{r_o}{\sin(135^\circ)}$$

Açıları ve Vertex noktalarını tespit etmek

Aşama 1 : Vertex noktalarını bulalım



$x_0 = r_o \times \cos(0^o)$

$y_0 = r_o \times \sin(0^o)$

$x_1 = r_i \times \cos(30^o)$

$y_1 = r_i \times \sin(30^o)$

$x_2 = r_o \times \cos(60^o)$

$y_2 = r_o \times \sin(60^o)$

$x_3 = r_i \times \cos(90^o)$

$y_3 = r_i \times \sin(90^o)$

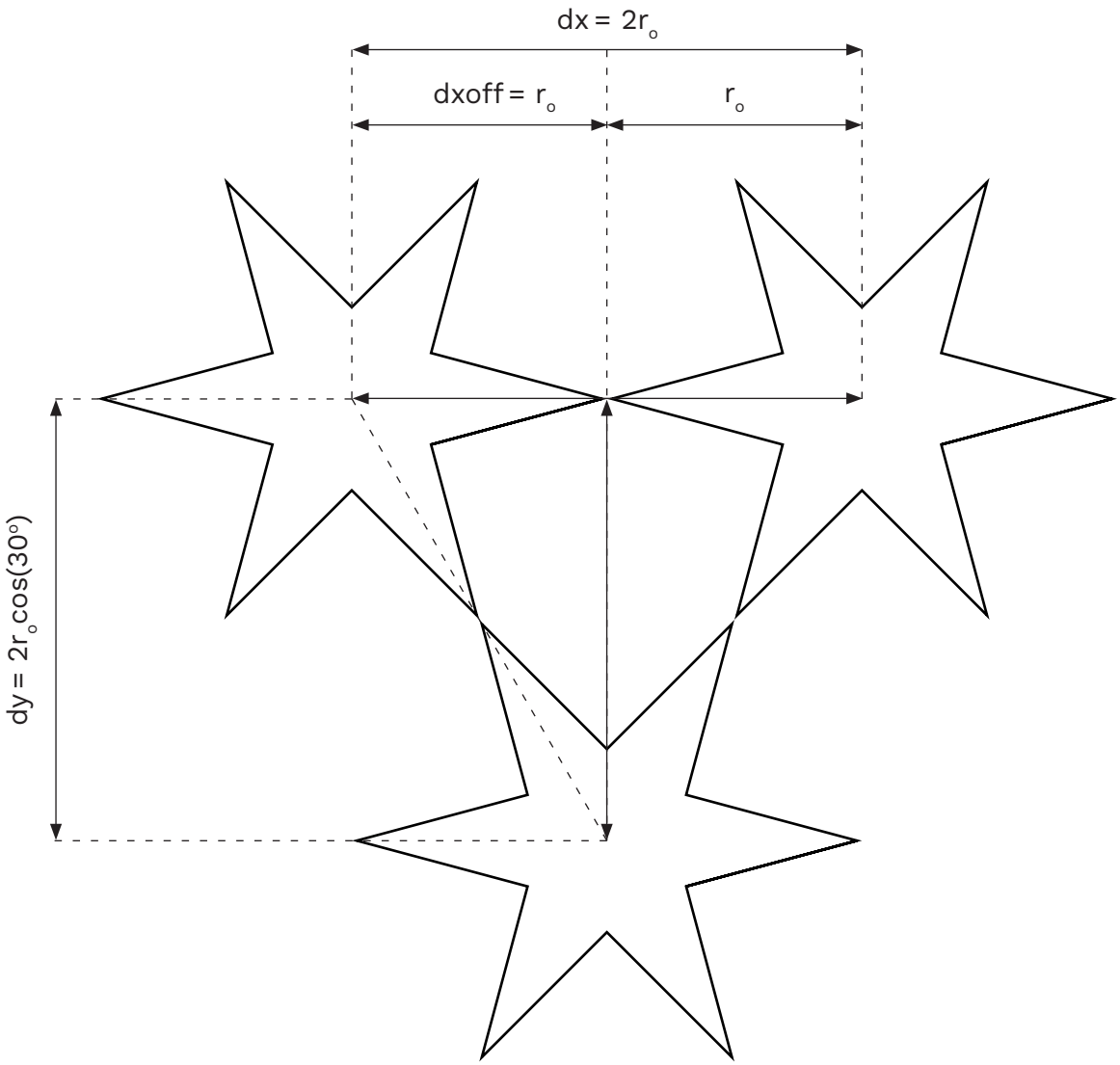
...

Motifi Oluşturmak

```
let a; //inner Radius
let b; //outer Radius

function setup() {
    createCanvas(400, 400);
    angleMode(DEGREES);
    noFill();
    noLoop();
    a = 48;
    b = a * (sin(135) / sin(15));
}

function draw() {
    let angle = 30;

    push();
        translate(width*0.5,height*0.5);
        beginShape();
        for (let i = 0; i < 12; i++) {
            let sx,sy;
            if(i%2==0){
                sx = cos(i*angle) * b;
                sy = sin(i*angle) * b;
            }else{
                sx = cos(i*angle) * a;
                sy = sin(i*angle) * a;
            }
        vertex(sx, sy);
        }
        endShape(CLOSE);
    pop();
}
```

Bezeme Yapısını İnceleyelim

Aşama 2 : Yerleştirmedeki dx, dy ve dxoff değerlerini hesaplamamız gerekiyor.



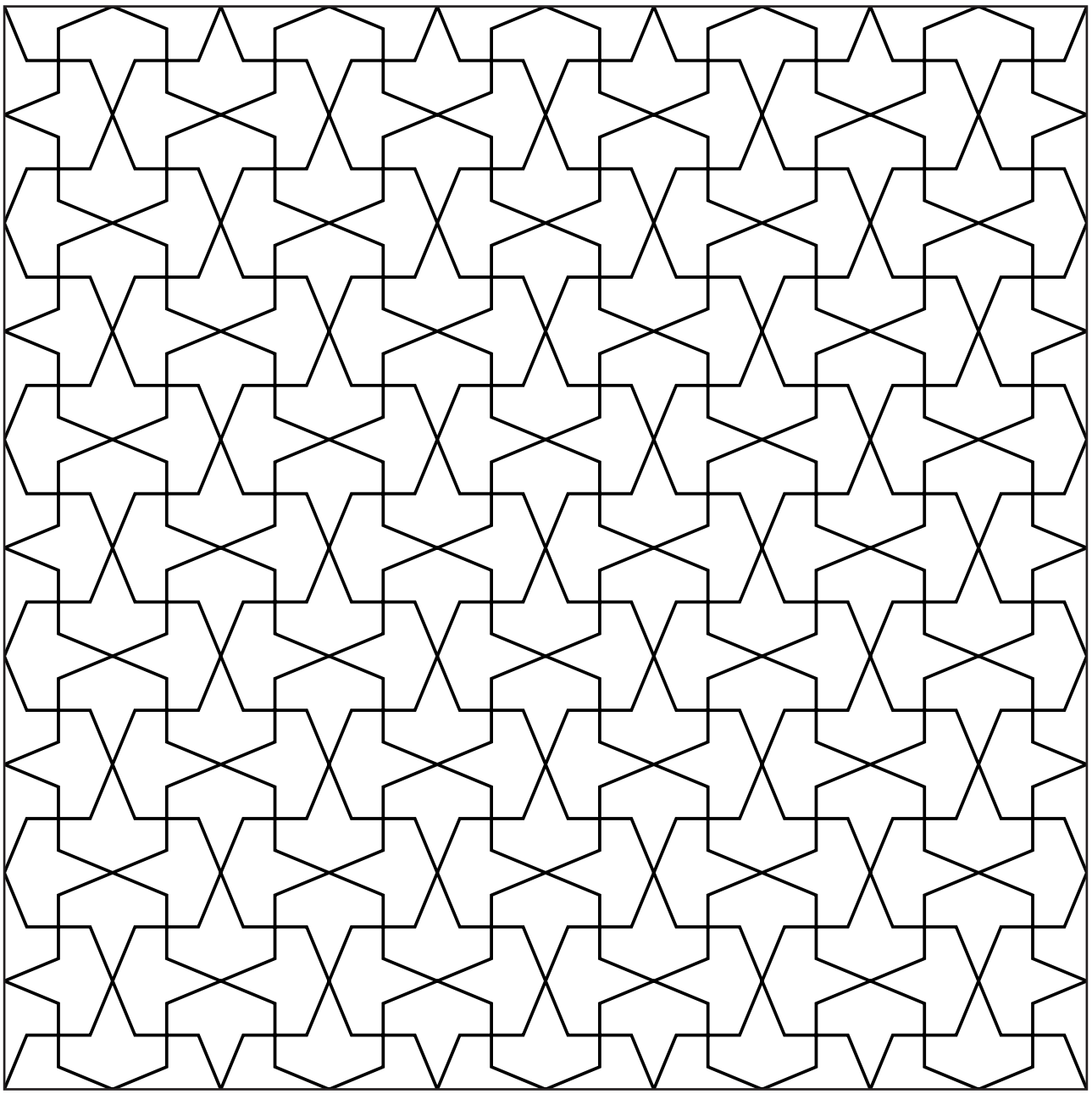
Bezeme Kodu

```
// Motif class
class Motif {
    constructor(r) {
        this.a = r; //inner Radius
        this.b = r * (sin(135) / sin(15)); //outer Radius
    }

    display() {
        let angle = 30;
        beginShape();
        for (let i = 0; i < 12; i++) {
            let sx, sy;
            if (i % 2 == 0) {
                sx = cos(i * angle) * this.b;
                sy = sin(i * angle) * this.b;
            } else {
                sx = cos(i * angle) * this.a;
                sy = sin(i * angle) * this.a;
            }
            vertex(sx, sy);
        }
        endShape(CLOSE);
    }
}

let a; //inner Radius
let b; //outer Radius
let dx, dy;
let nRow;
let nCol;

function setup() {
    createCanvas(800, 800);
    angleMode(DEGREES);
    noFill();
    noLoop();
    a = 16;
    b = a * (sin(135) / sin(15));
    dx = 2 * b;
    dy = 2 * b * cos(30);
    //approximate the nRow and nCol values
    nRow = ceil(height / dy);
    nCol = ceil(width / dx);
}
```

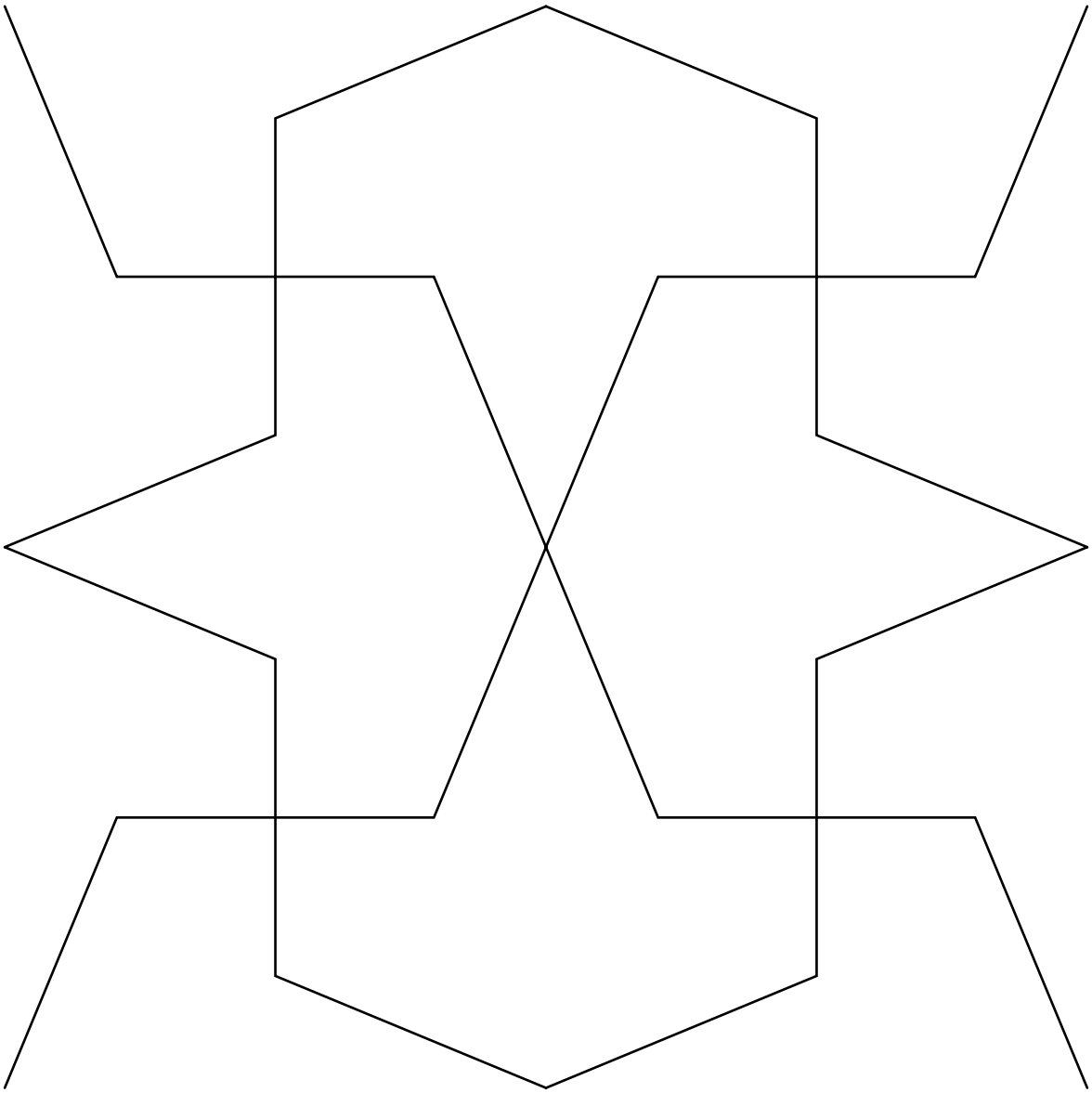
```
function draw() {
    let motif = new Motif(a);
    for (let r = 0; r < nRow; r++) {
        for (let c = 0; c < nCol; c++) {
            push();
                if (r % 2 == 0) {
                    //rows 0,2,4,6
                    translate(c * dx, r * dy);
                } else {
                    //rows 1,3,5,7
                    translate(c * dx + b, r * dy);
                }
                motif.display();
            pop();
        }
    }
}
```

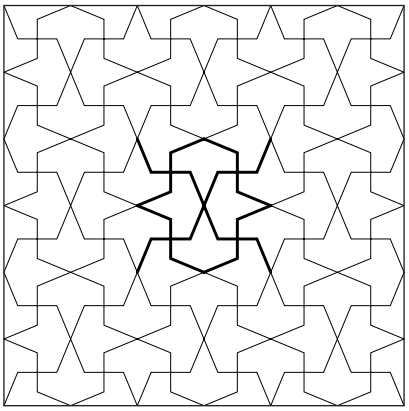
Geometrik Deseni Kodlamak
Örnek 5

Aşağıdaki deseni inceleyin ve bu deseni oluşturan temel görsel bileşeni bulmaya çalışın

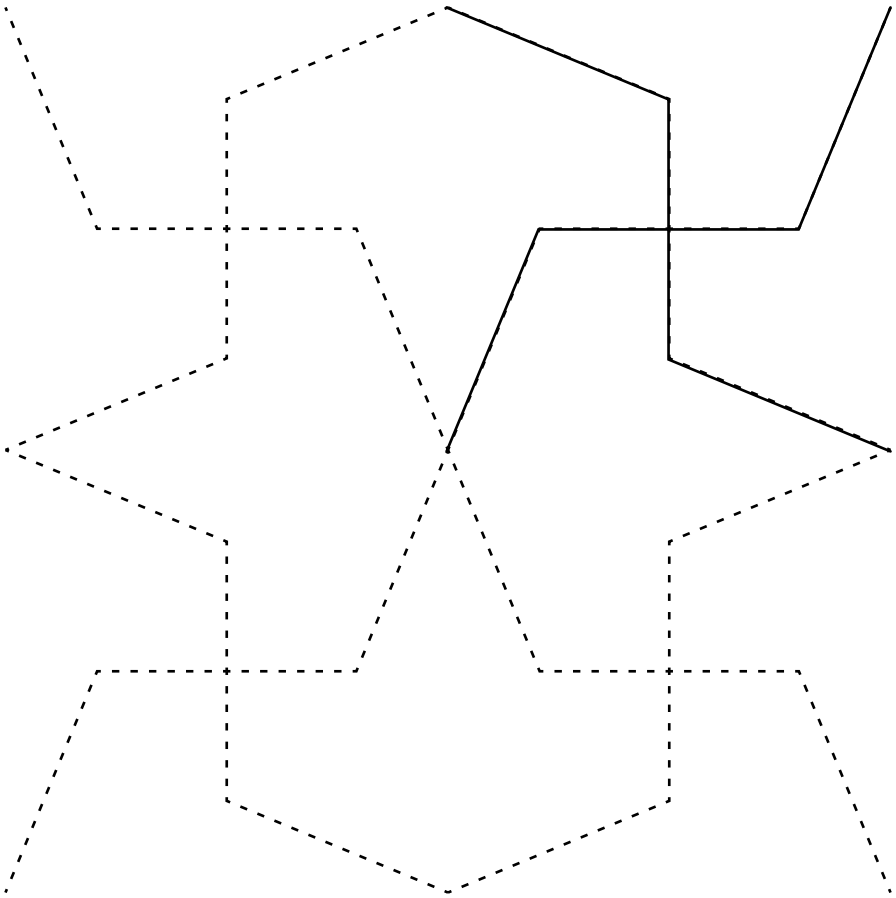


Motif



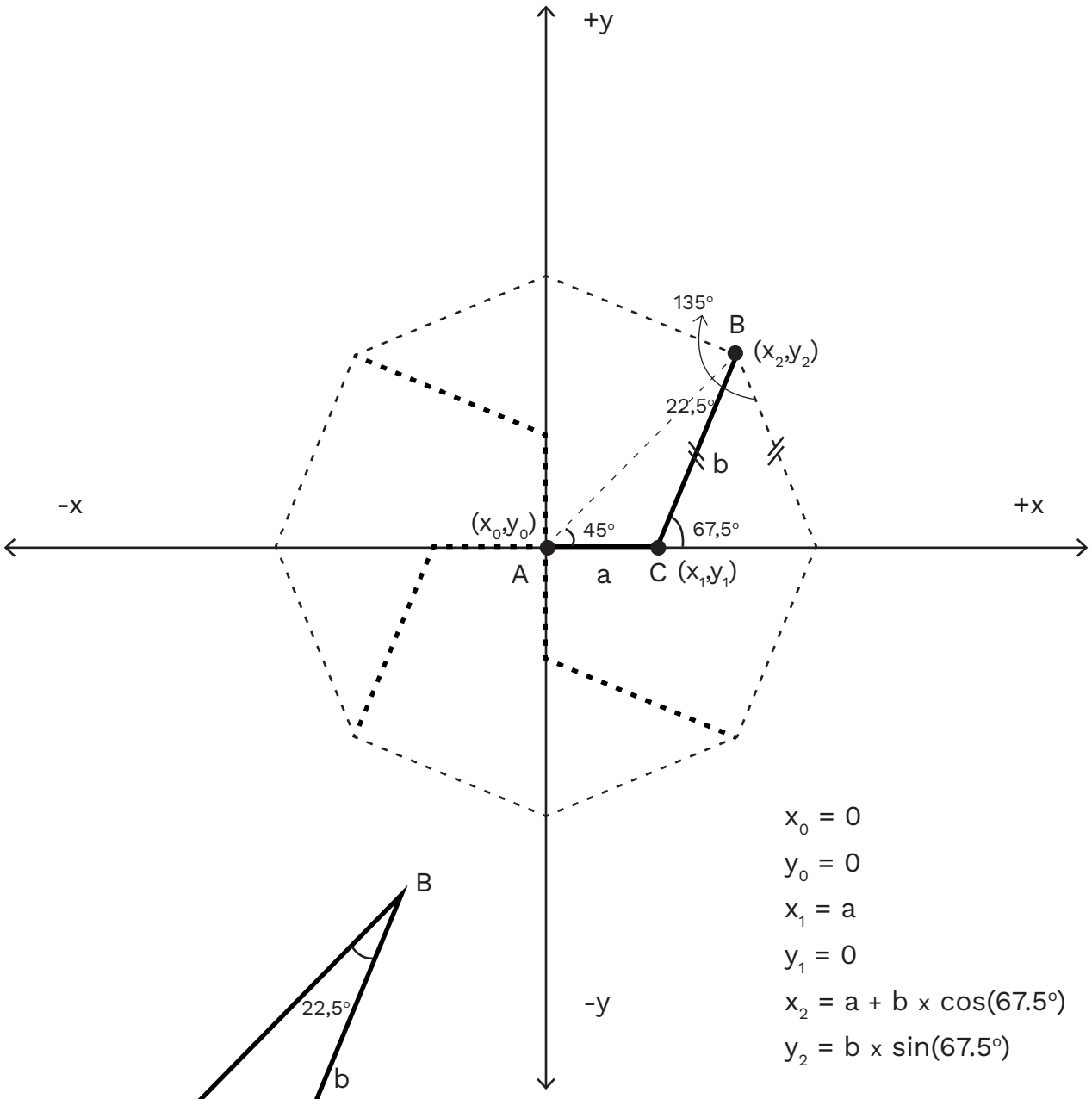


Temel Görsel Bileşini İnceleyelim



Açıları ve Vertex noktalarını tespit etmek

Aşama 1 : Vertex noktalarını bulalım



$x_0 = 0$

$y_0 = 0$

$x_1 = a$

$y_1 = 0$

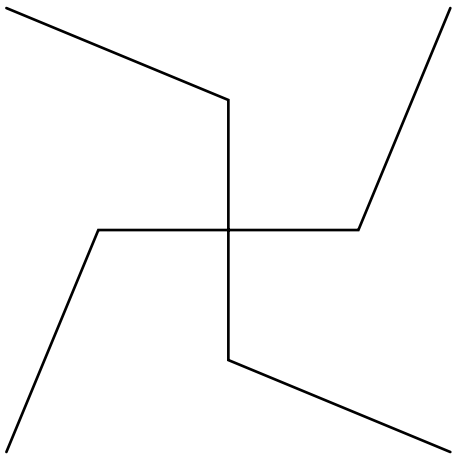
$x_2 = a + b \times \cos(67.5^\circ)$

$y_2 = b \times \sin(67.5^\circ)$

$$\frac{a}{\sin(22.5^\circ)} = \frac{b}{\sin(45^\circ)}$$

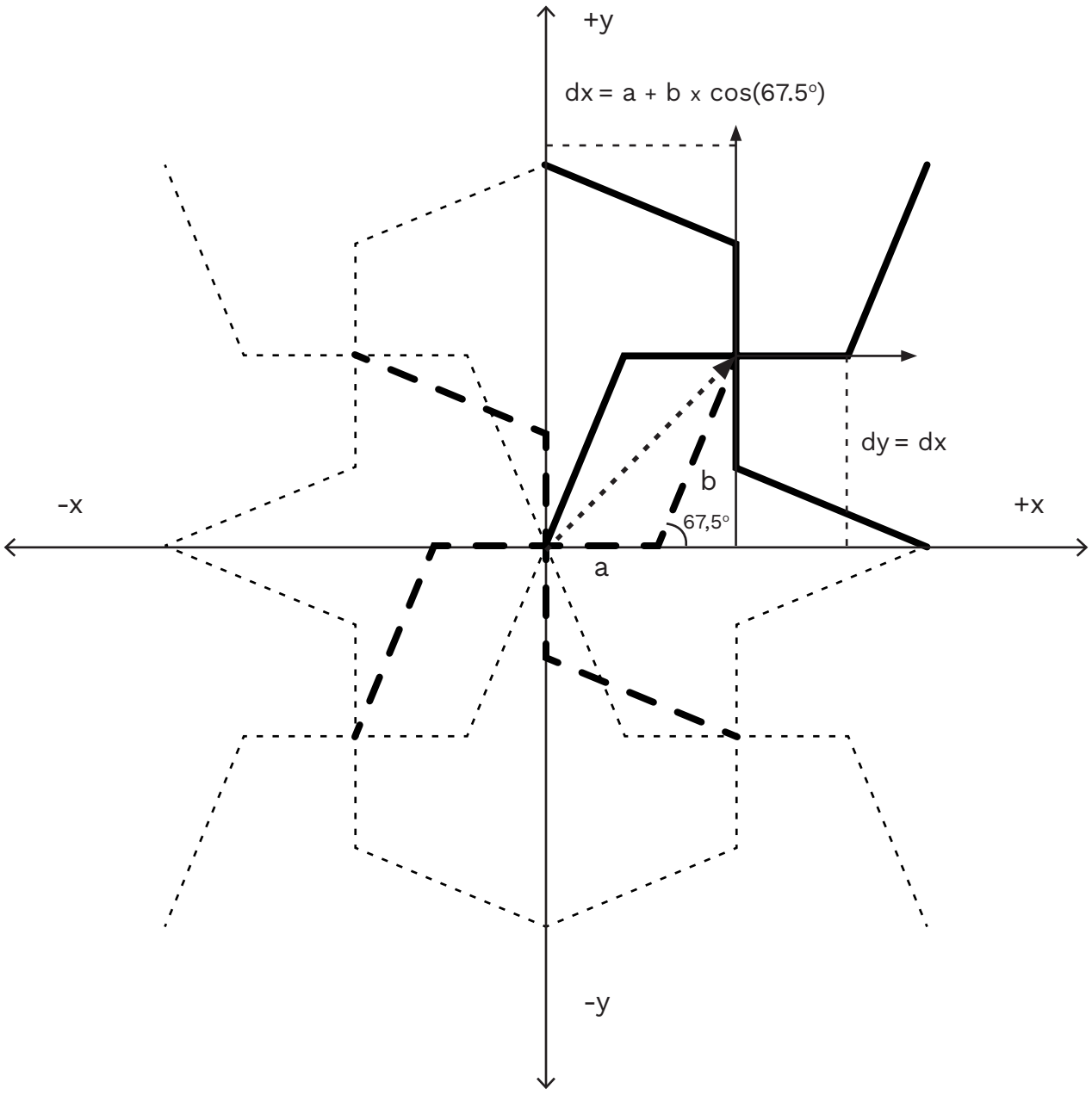
Motifi Oluşturmak

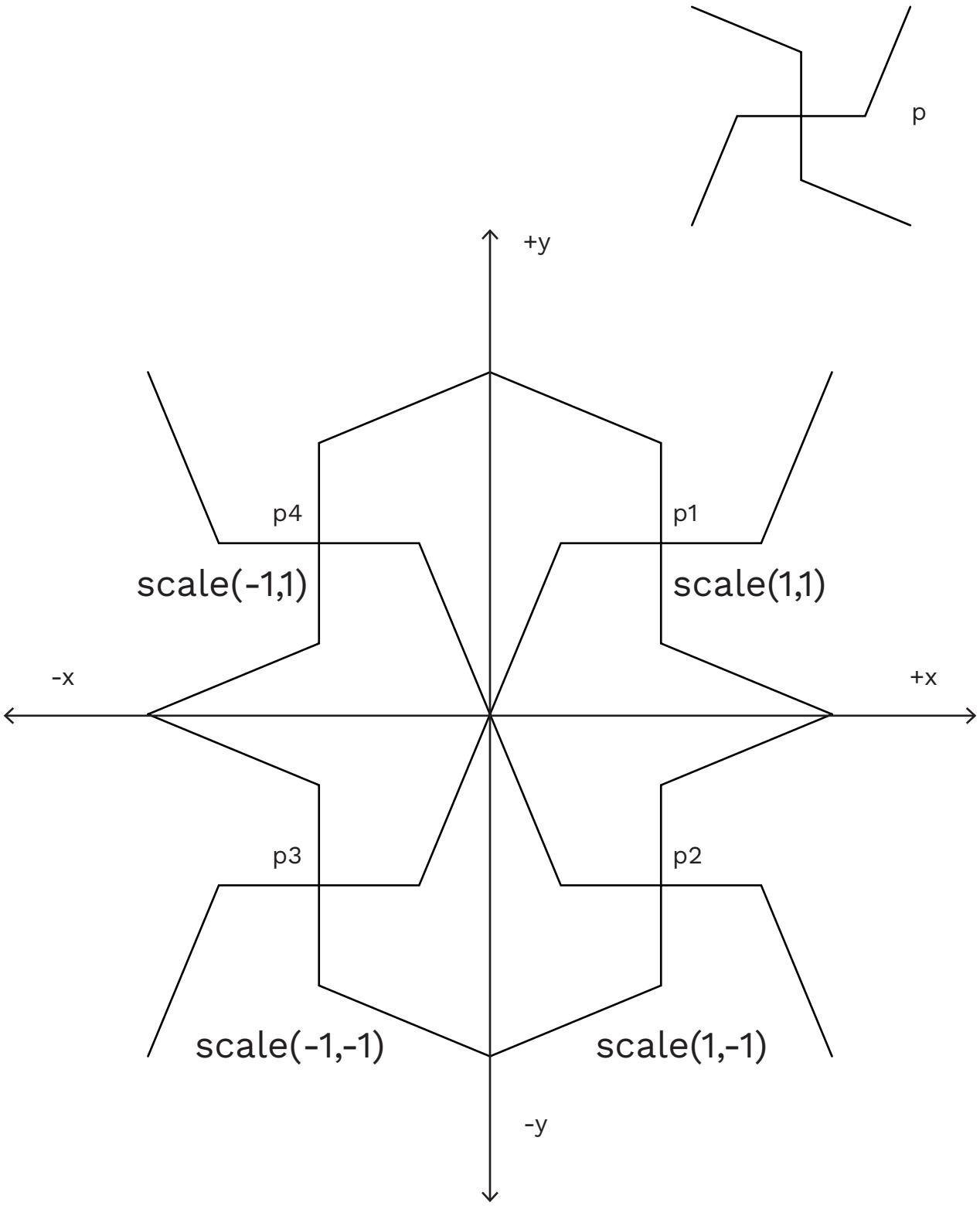
```
//scale factor
let a,b;
function setup() {
    createCanvas(400, 400);
    angleMode(DEGREES);
    noFill();
    a = 40;
    b = a * (sin(45) / sin(22.5));
}

function draw() {
    noFill();
    let x0,y0,x1,y1,x2,y2;
    push();
        translate(width*0.5,height*0.5);
        for(let i=0;i<4;i++){
            rotate(i*90);
            beginShape();
            x0 = 0;
            y0 = 0;
            x1 = a;
            y1 = 0;
            x2 = a + b * cos(67.5);
            y2 = b * sin(67.5);
            vertex(x0,-y0);
            vertex(x1,-y1);
            vertex(x2,-y2);
            endShape();
        }
    pop();
}
```



Açıları ve Vertex noktalarını tespit etmek

Aşama 2 : Çeyreği kullanarak tam motifi oluşturmamız gerekiyor





p2, p1'in yatay eksene göre ayna yansımasıdır
p3, p1'in yatay ve dikey eksene göre ayna yansımasıdır
p4, p1'in dikey eksene göre ayna yansımasıdır

yansımalar oluşturmak için ölçek dönüştürme "scale" işlevini kullanıyoruz

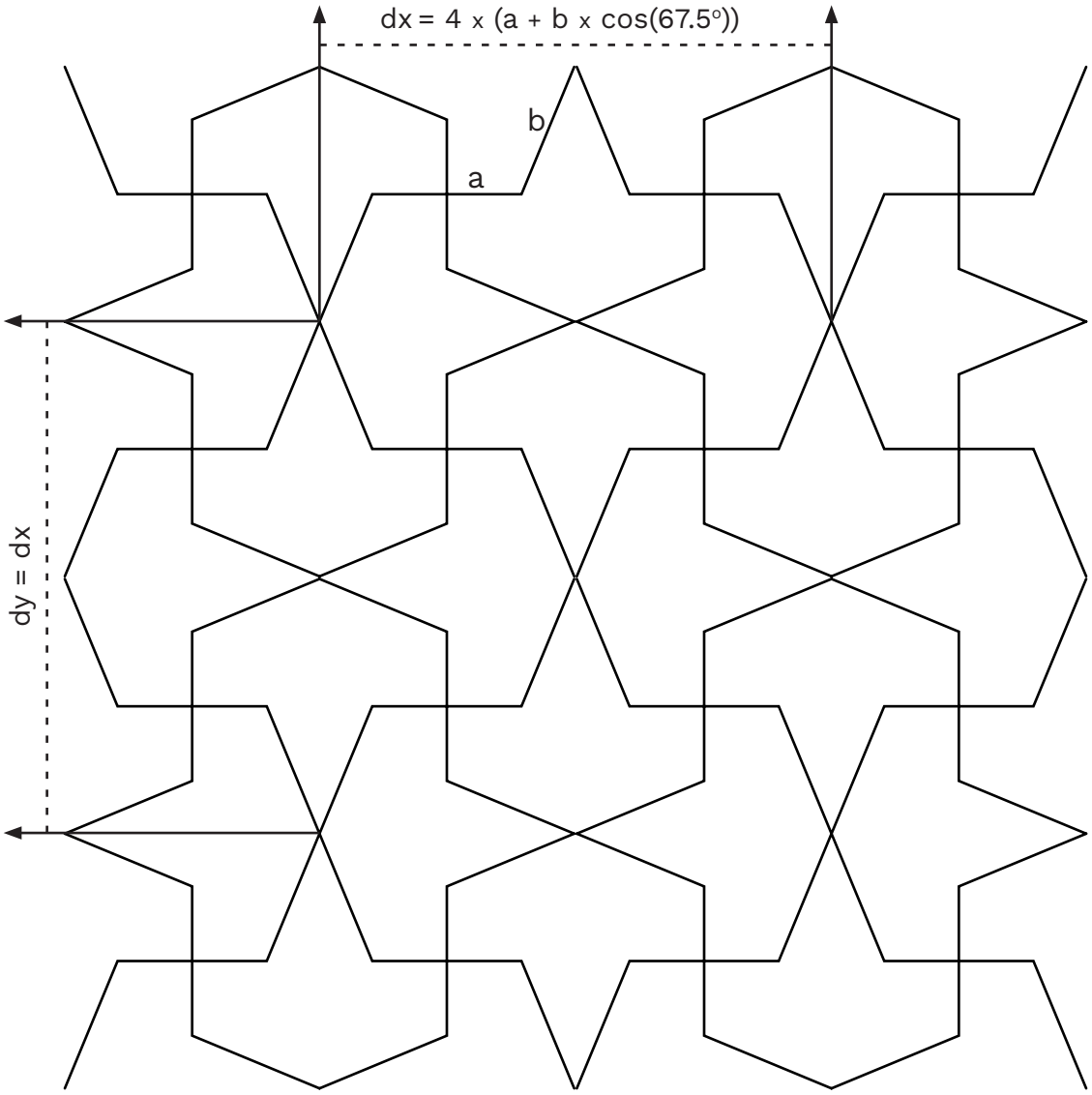
Motifi Oluşturmak

```
//scale factor
let a, b;
function setup() {
    createCanvas(400, 400);
    angleMode(DEGREES);
    noFill();
    a = 40;
    b = a * (sin(45) / sin(22.5));
}

function draw() {
    noFill();
    let x0, y0, x1, y1, x2, y2;
    let dx = a + b * cos(67.5);
    let dy = dx;
    push();
        translate(width * 0.5, height * 0.5);
        for (let k = 0; k < 4; k++) {
            push();
                //mirroring
                switch (k) {
                    case 0:
                        scale(1, 1);
                        break;
                    case 1:
                        scale(1, -1);
                        break;
                    case 2:
                        scale(-1, -1);
                        break;
                    case 3:
                        scale(-1, 1);
                        break;
                    default:
                    //
                }
```

```
            translate(dx, -dy);
            //quarter shape
            for (let i = 0; i < 4; i++) {
                rotate(i * 90);
                beginShape();
                x0 = 0;
                y0 = 0;
                x1 = a;
                y1 = 0;
                x2 = a + b * cos(67.5);
                y2 = b * sin(67.5);
                vertex(x0, -y0);
                vertex(x1, -y1);
                vertex(x2, -y2);
                endShape();
            }
        pop();
    }
  pop();
}
```

Bezeme Yapısını İnceleyelim

Aşama 3 : Yerleştirmedeki dx ve dy değerlerini hesaplamamız gerekiyor.



Bezeme Kodu

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

    display() {
        let x0, y0, x1, y1, x2, y2;
        let b = this.a * (sin(45) / sin(22.5));
        let dx = this.a + b * cos(67.5);
        let dy = dx;

        for (let k = 0; k < 4; k++) {
            push();
                //mirroring
                switch (k) {
                    case 0:
                        scale(1, 1);
                        break;
                    case 1:
                        scale(1, -1);
                        break;
                    case 2:
                        scale(-1, -1);
                        break;
                    case 3:
                        scale(-1, 1);
                        break;
                    default:
                    //
                }
                translate(dx, -dy);
                //quarter shape
                for (let i = 0; i < 4; i++) {
                    rotate(i * 90);
                    beginShape();
                    x0 = 0;
                    y0 = 0;
                    x1 = a;
                    y1 = 0;
                    x2 = a + b * cos(67.5);
                    y2 = b * sin(67.5);
                    vertex(x0, -y0);
                    vertex(x1, -y1);
                    vertex(x2, -y2);
                    endShape();
                }
```

```
            pop();
        }
    }
}

let a = 20;
let xOff, yOff;
let nRow;
let nCol;
let motif = new Motif(a);

function setup() {
    createCanvas(800, 800);
    angleMode(DEGREES);
    noLoop();
    noFill();

    let b = a * (sin(45) / sin(22.5));
    xOff = 4 * (a + b * cos(67.5));
    yOff = xOff;

    //approximate the nRow and nCol values
    nRow = 1 + ceil(height / xOff);
    nCol = 1 + ceil(width / yOff);
}

function draw() {
    push();
        for (let r = 0; r < nRow; r++) {
            for (let c = 0; c < nCol; c++) {
                push();
                    translate(xOff * c, yOff * r);
                    motif.display();
                pop();
            }
        }
    pop();
}
```