



**Bugün:**  
**04.Eylül.2025**

Geometri Desenleri

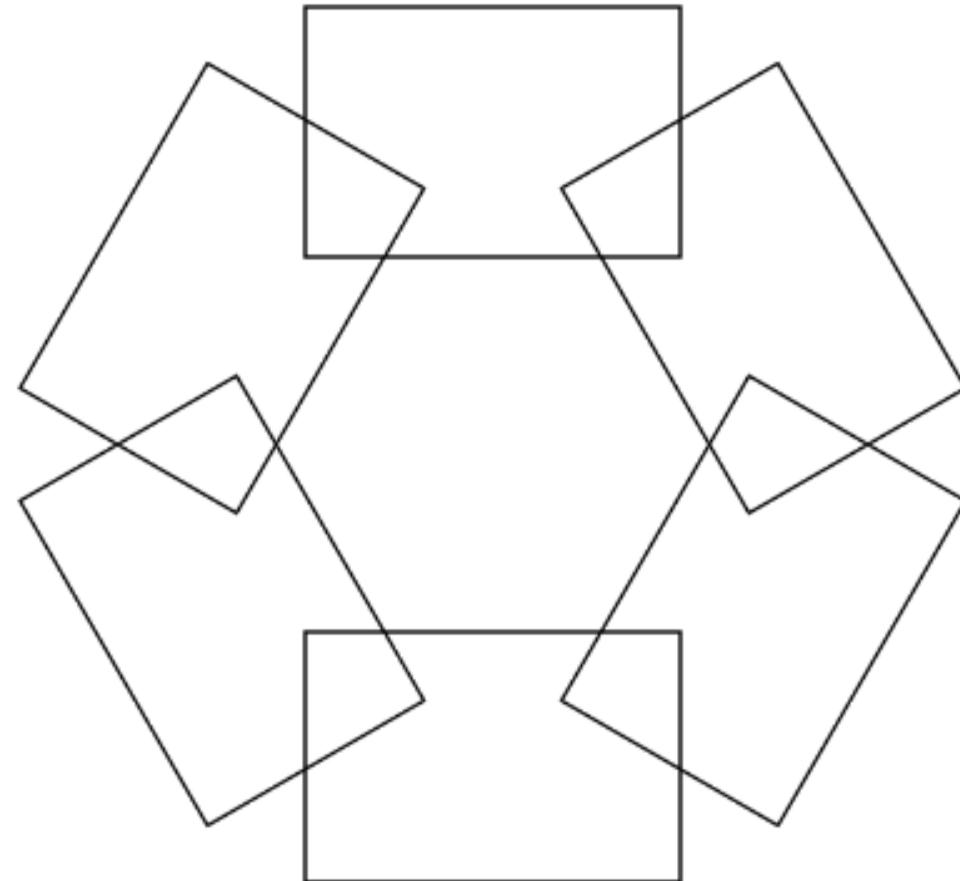
**Fig. 6.2** Carved masonry and stone relief, Agra Fort, Agra. See also (Fig. 14.3a). This example is dated 1573, with earlier examples in Iran and Cairo. This pattern is unusual in having a 6-way intersection (Patterns in Islamic Art web site 2017, IND 0332)





```
function setup() {  
  createCanvas(400, 400);  
  rectMode(CENTER);  
  angleMode(DEGREES);  
  noFill();  
}  
  
function draw() {
```

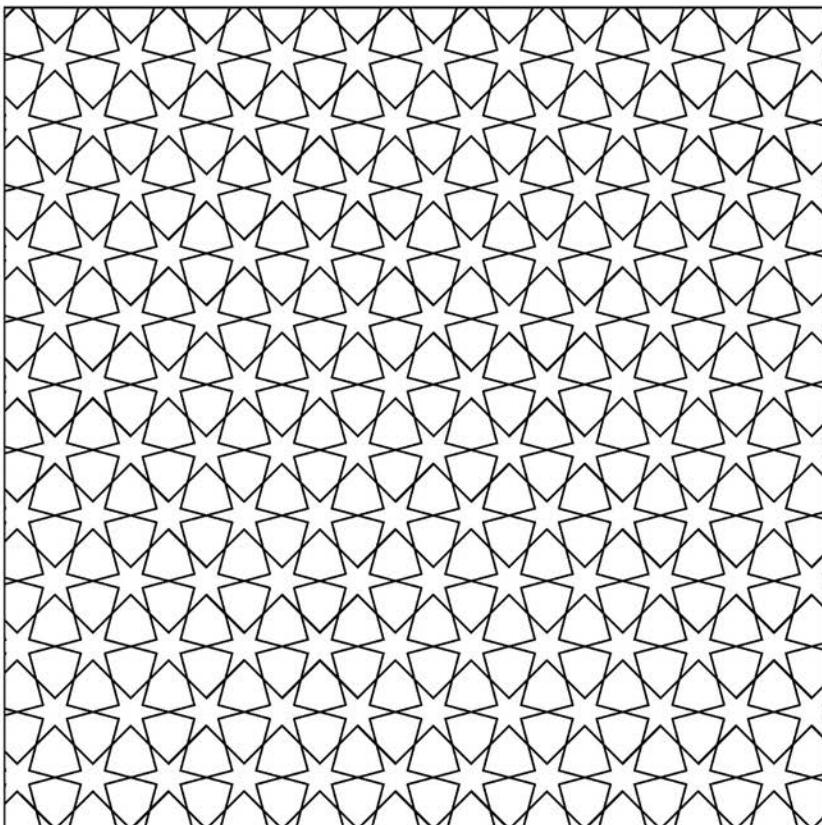
```
background(255);  
push();  
translate(width*0.5,height*0.5);  
rotate(90);  
for(let i = 0; i<6 ; i++){  
  push();  
  rotate(i*60);  
  translate(100,0);  
  rect(0,0,80,120);  
  pop();  
}  
pop();  
}
```



## **Geometrik Deseni Kodlamak**

### **Örnek 2: Eşrefoğlu Camii, Beyşehir**

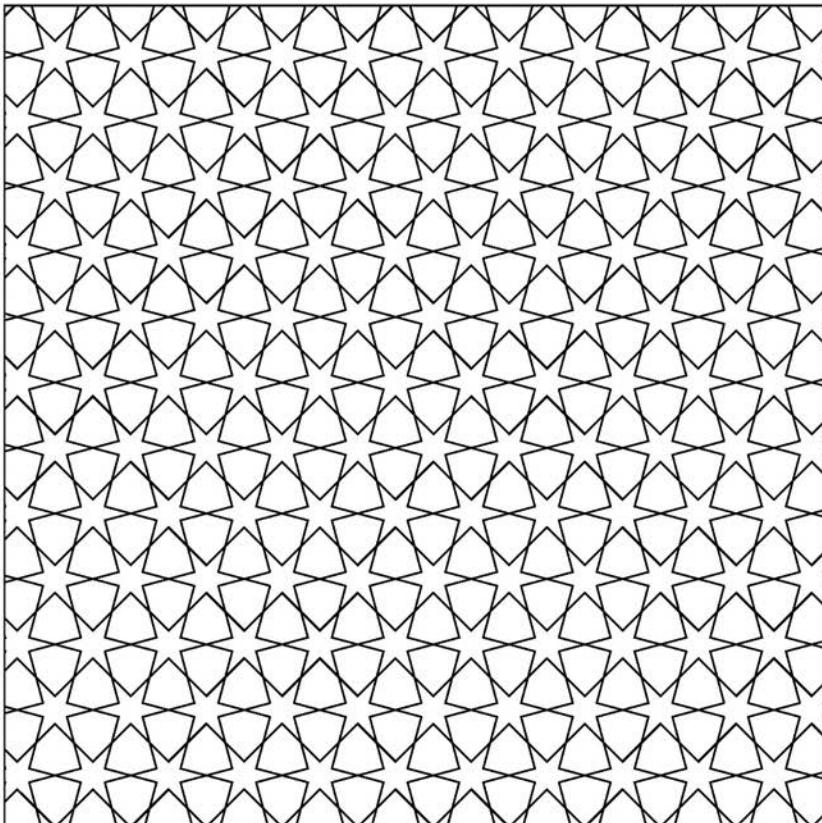
Aşağıdaki deseni inceleyin ve bu deseni oluşturan temel görsel bileşeni bulmaya çalışın



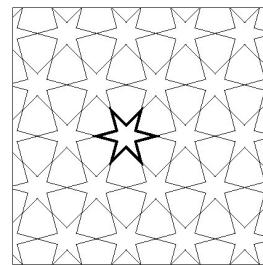
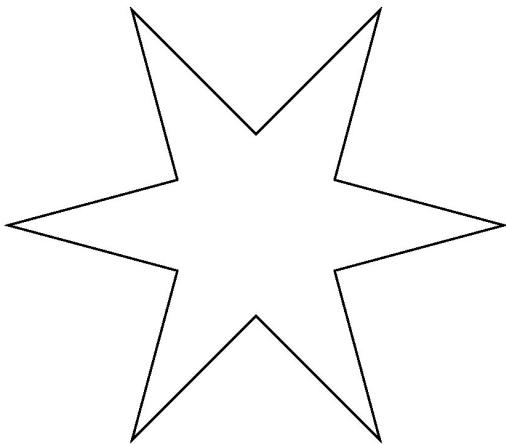
## **Geometrik Deseni Kodlamak**

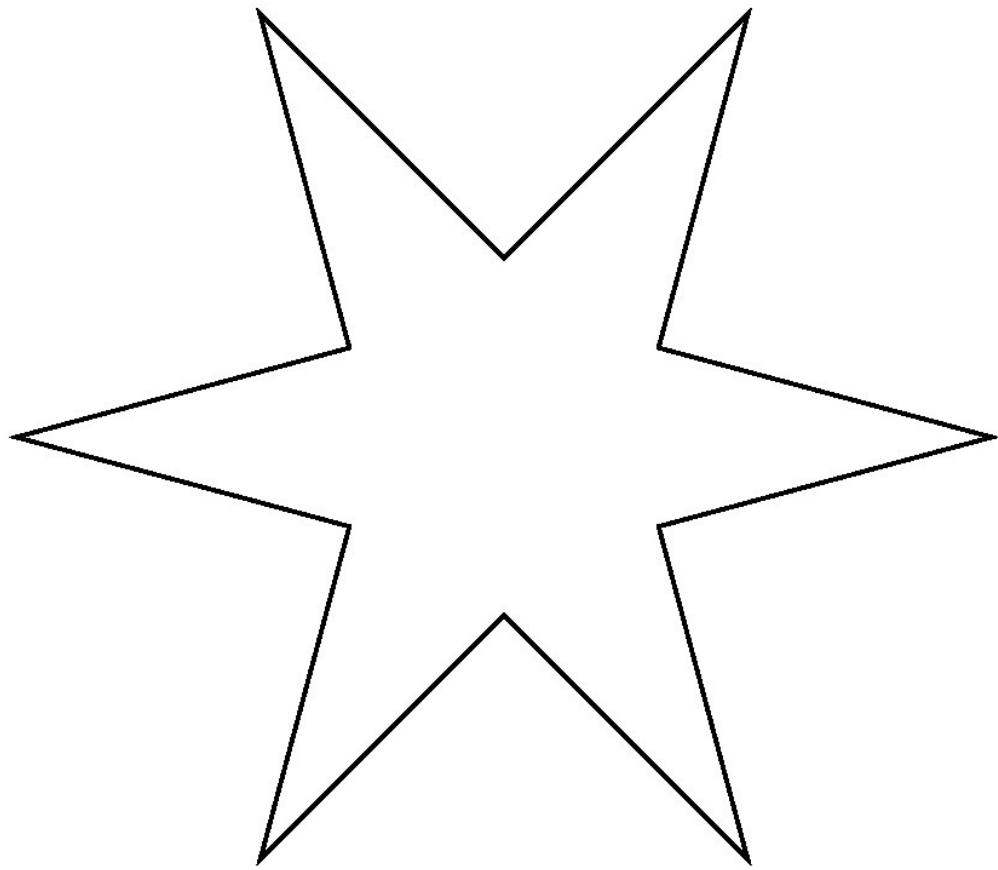
### **Örnek 2: Eşrefoğlu Camii, Beyşehir**

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

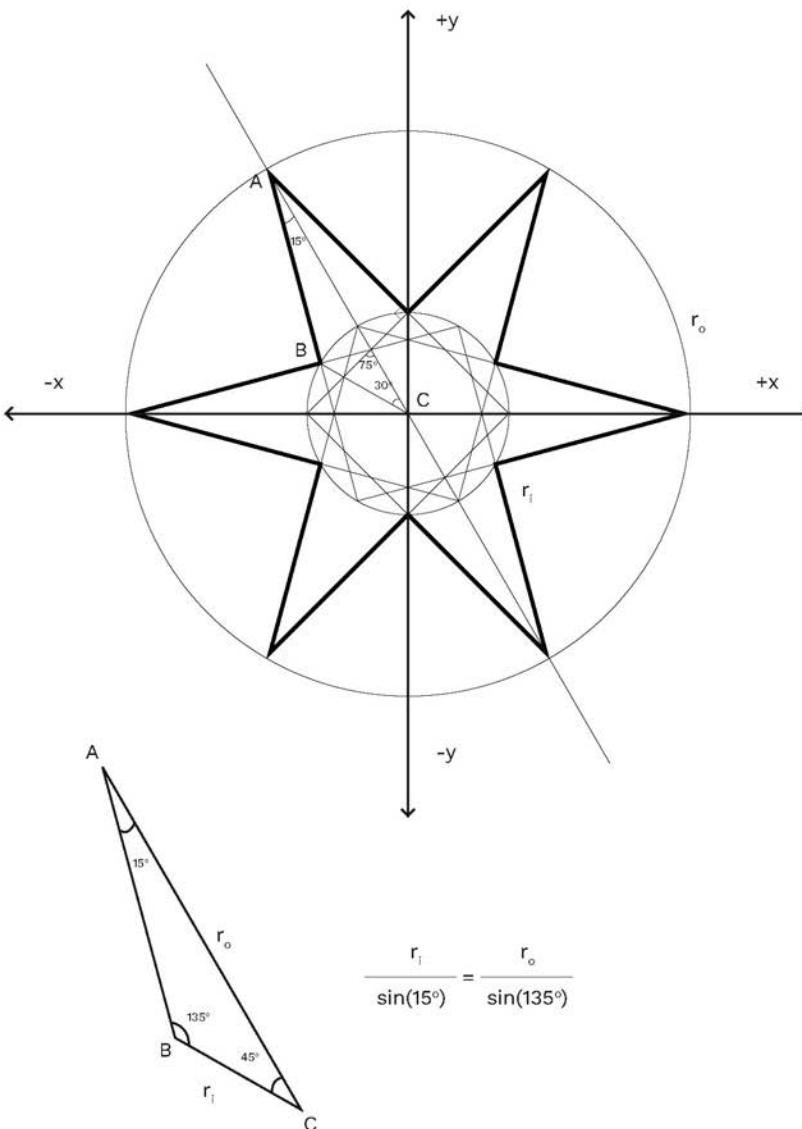


Motif



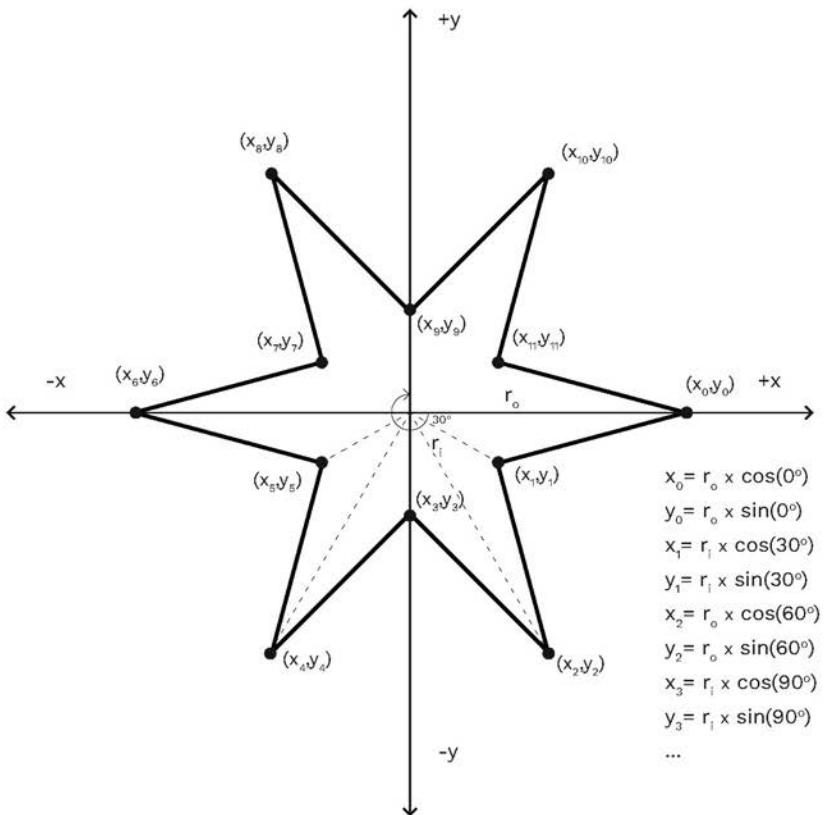


## Temel Görsel Bileşini İnceleyelim



## Açıları ve Vertex noktalarını tespit etmek

Aşama 1 : Vertex noktalarını bulalım

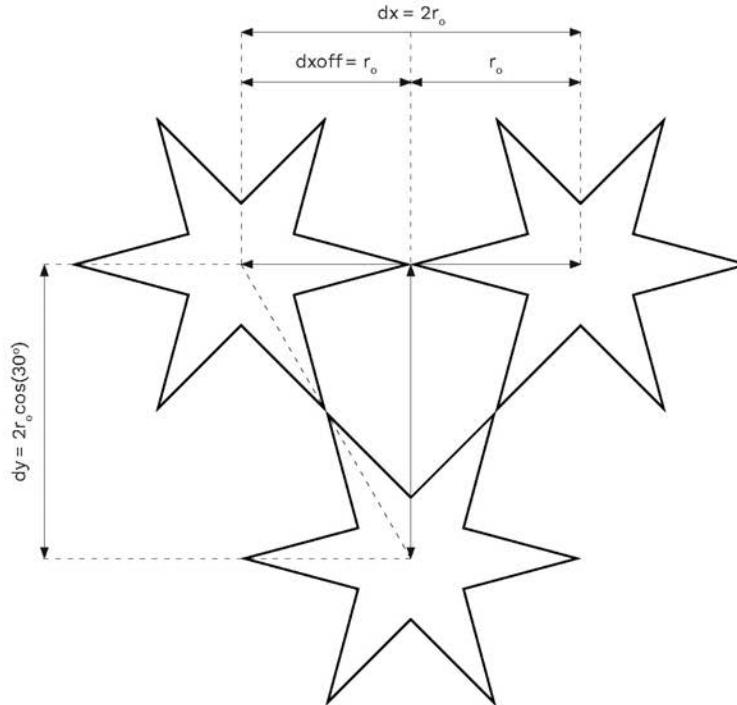


## Motifi Oluşturmak

```
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  $dx_{off}$  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);  
}

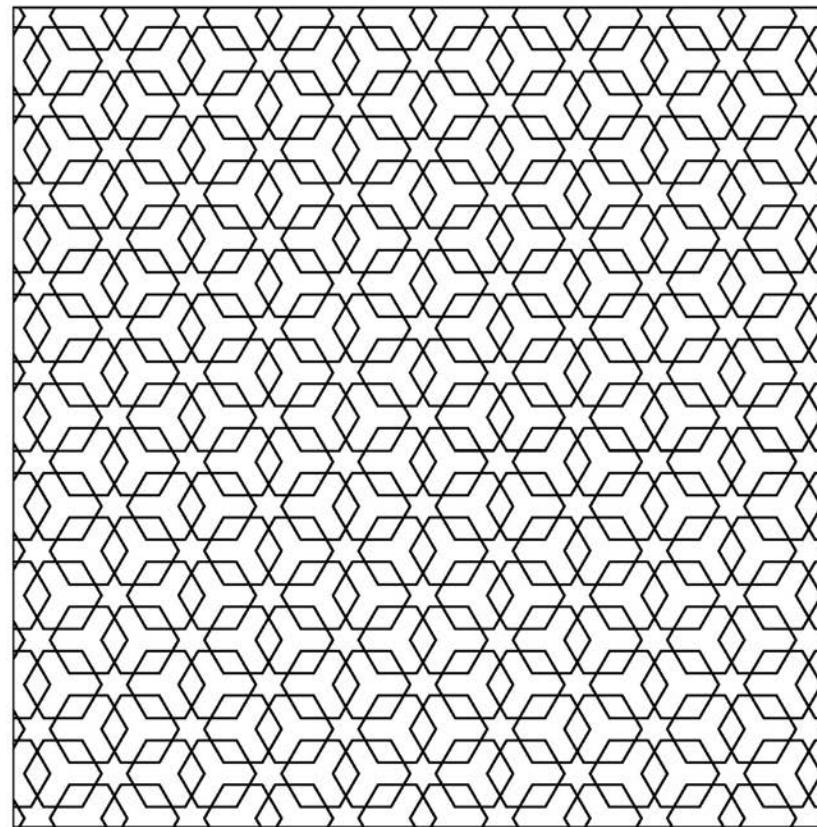
```
function draw() {
    let motif = new Motif(a);
    for (let r = 0; r < nRow; r++) {
        for (let c = 0; c < nCol; c++) {
            push();
            if (r % 2 == 0) {
                //rows 0,2,4,6
                translate(c * dx, r * dy);
            } else {
                //rows 1,3,5,7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



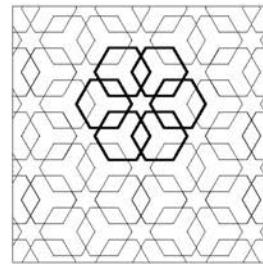
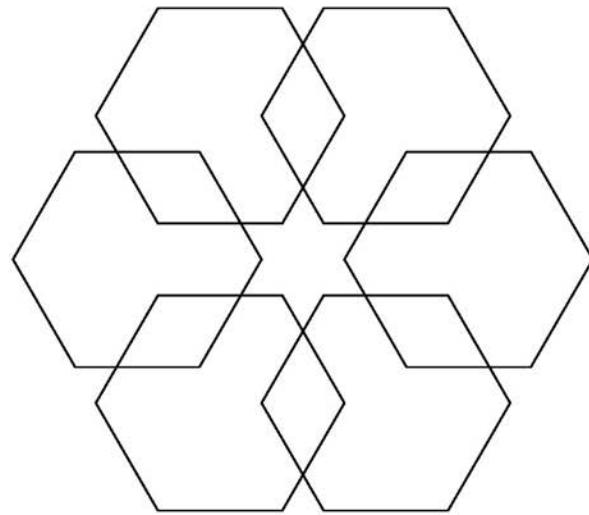
60. Mosaic tile dado panel in the vestibule behind the entrance iwan, Masjid-i Jamī', Yazd, fifteenth century. Photo: Courtesy Harvard University, Fogg Art Museum, Fine Arts Library, Visual Collections.

**Geometrik Deseni Kodlamak**  
**Örnek 3: Eşrefoğlu Camii, Beyşehir**

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

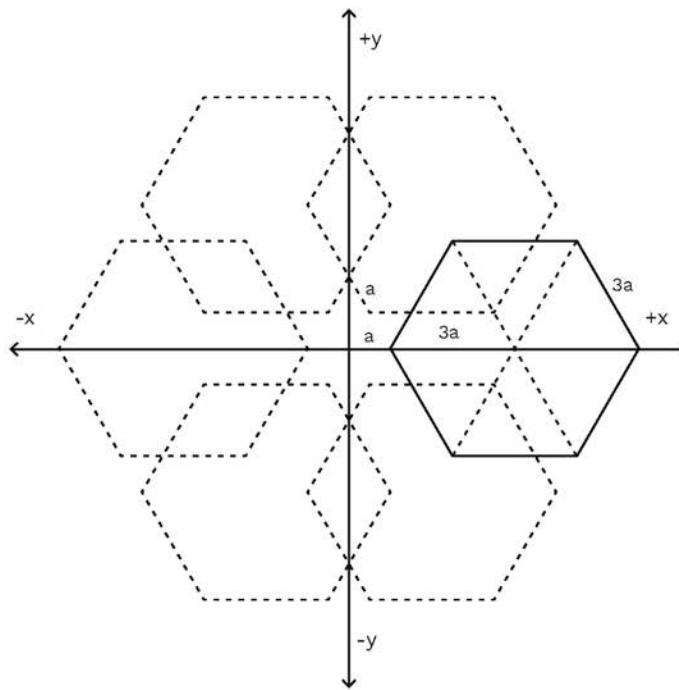


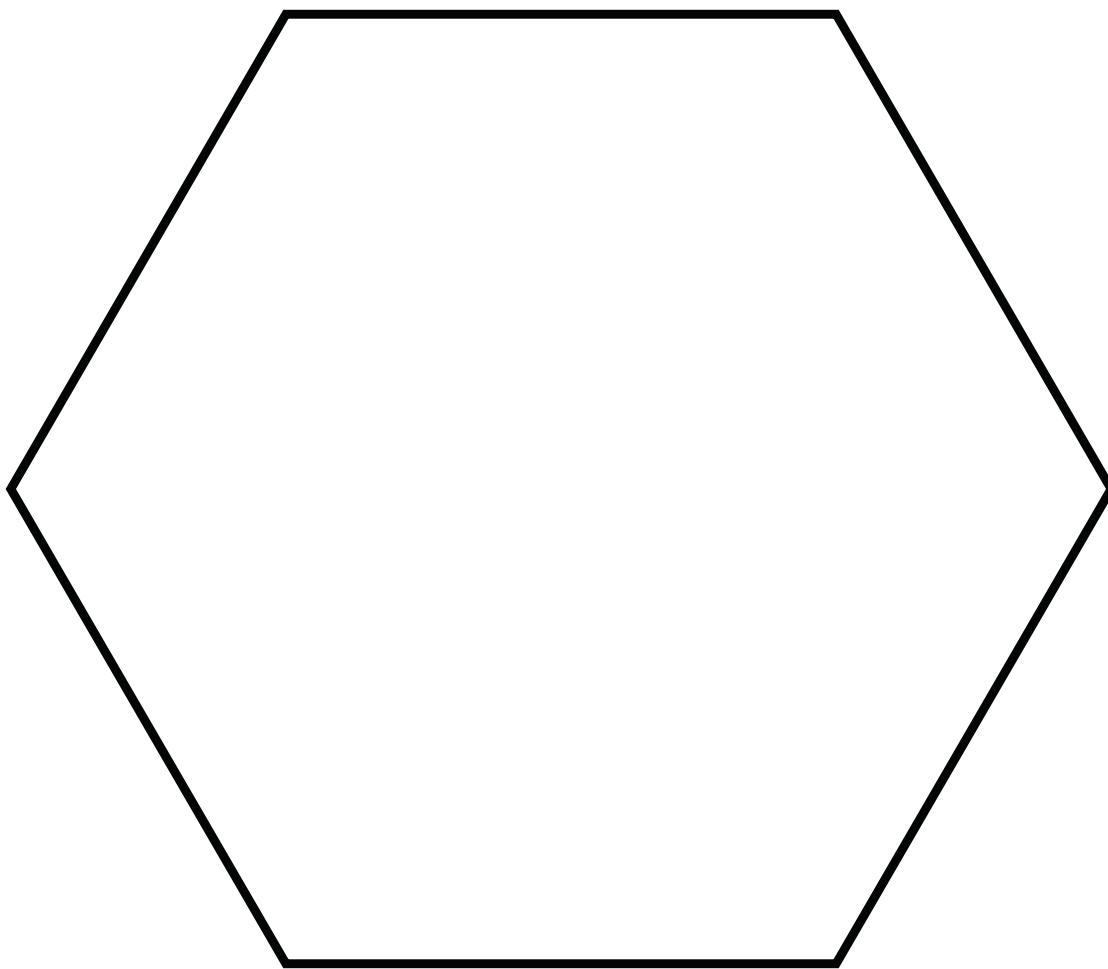
Motif



## Temel Görsel Bileşini İnceleyelim

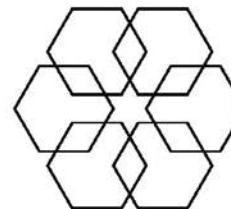
Aşama 1 : Orijin etrafında dönen altı tekrarlı, eşit kenarlı bir altıgen olduğunu gözlemliyoruz.





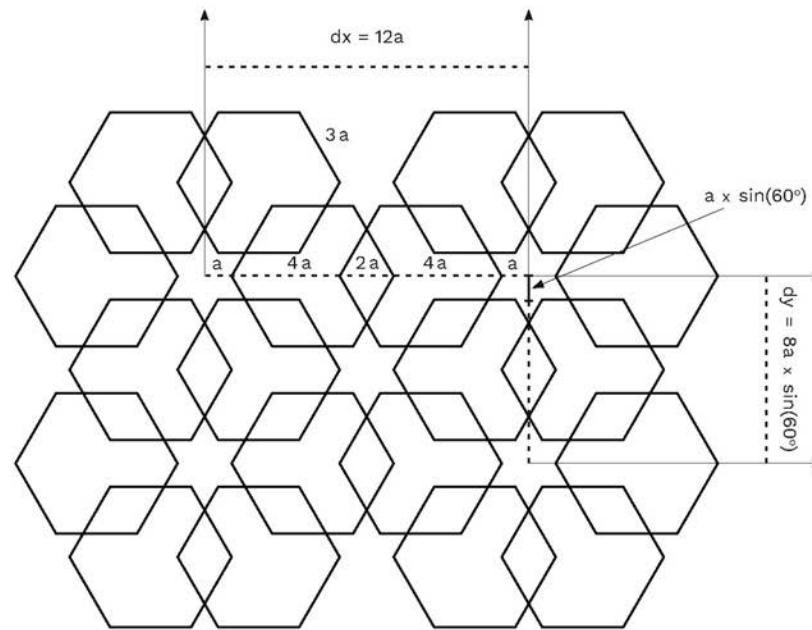
## Motifi Oluşturmak

```
//scale factor  
let a = 20;  
  
function setup() {  
    createCanvas(400, 400);  
    angleMode(DEGREES);  
    noFill();  
    noLoop();  
}  
  
function draw() {  
    push();  
    translate(width*0.5,height*0.5);  
    for(let i=0;i<6;i++){  
        push();  
        rotate(60*i);  
        translate(4*a,0);  
        beginShape();  
        //hexagon  
        for(let k=0;k<6;k++){  
            let x = 3 * a * cos(k*60);  
            let y = 3 * a * sin(k*60);  
            vertex(x,y);  
        }  
        endShape(CLOSE);  
        pop();  
    }  
    pop();  
}
```



## Bezeme Yapısını İnceleyelim

Aşama 2 : Yerleştirmedeki  $dx$  ve  $dy$  değerlerini hesaplamamız gerekiyor.



## Bezeme Kodu

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

    display() {
        for (let i = 0; i < 6; i++) {
            push();
            rotate(60 * i);
            translate(4 * this.a, 0);
            beginShape();
            //hexagon
            for (let k = 0; k < 6; k++) {
                let x = 3 * this.a * cos(k * 60);
                let y = 3 * this.a * sin(k * 60);
                vertex(x, y);
            }
            endShape(CLOSE);
            pop();
        }
    }
}

//scale factor
let a = 18;

let motif = new Motif(a);
let nRow;
let nCol;
let dx, dy;

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

    dx = 12 * a;
    dy = 8 * a * sin(60);
    //approximate the nRow and nCol values
    nRow = ceil(height / dy);
    nCol = 1+ceil(width / dx);
}
```

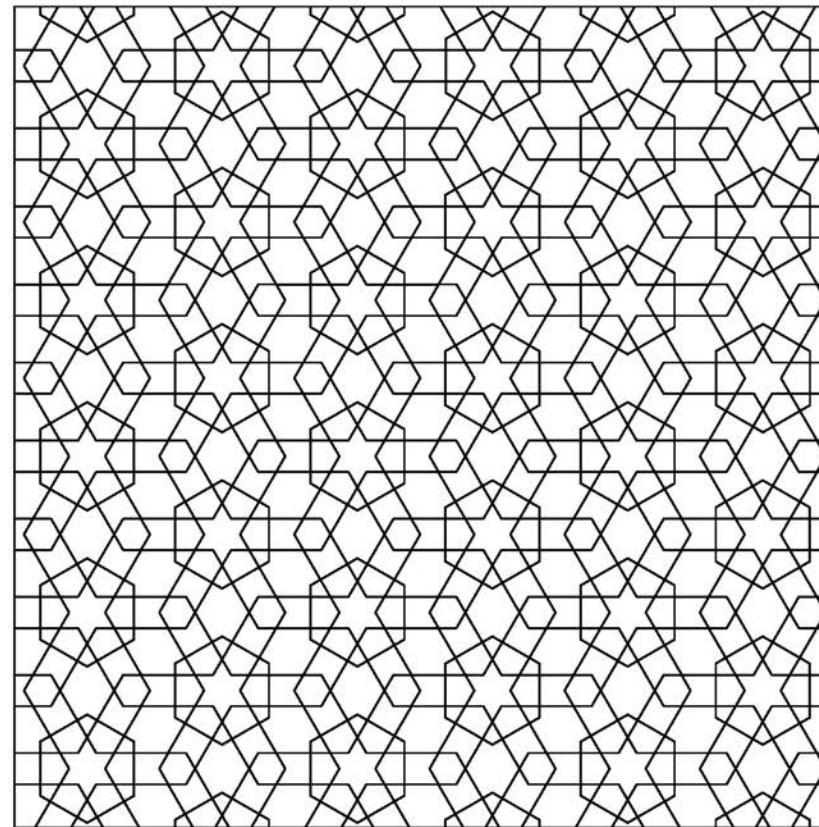
## Bezeme Kodu

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

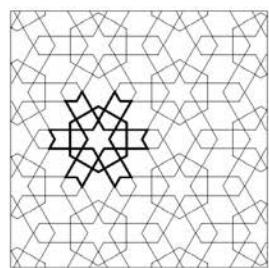
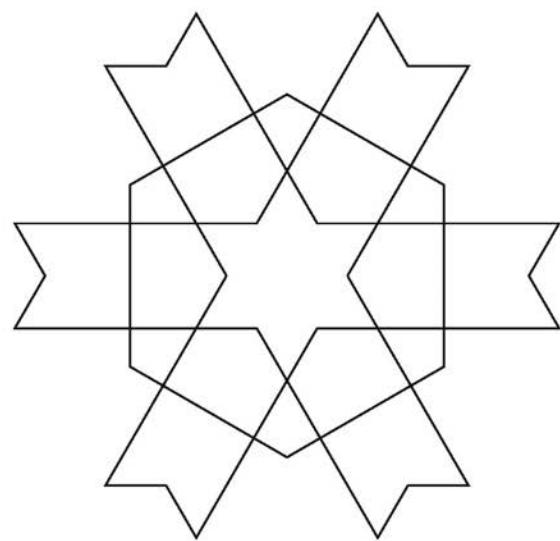
## **Geometrik Deseni Kodlamak**

### **Örnek 4**

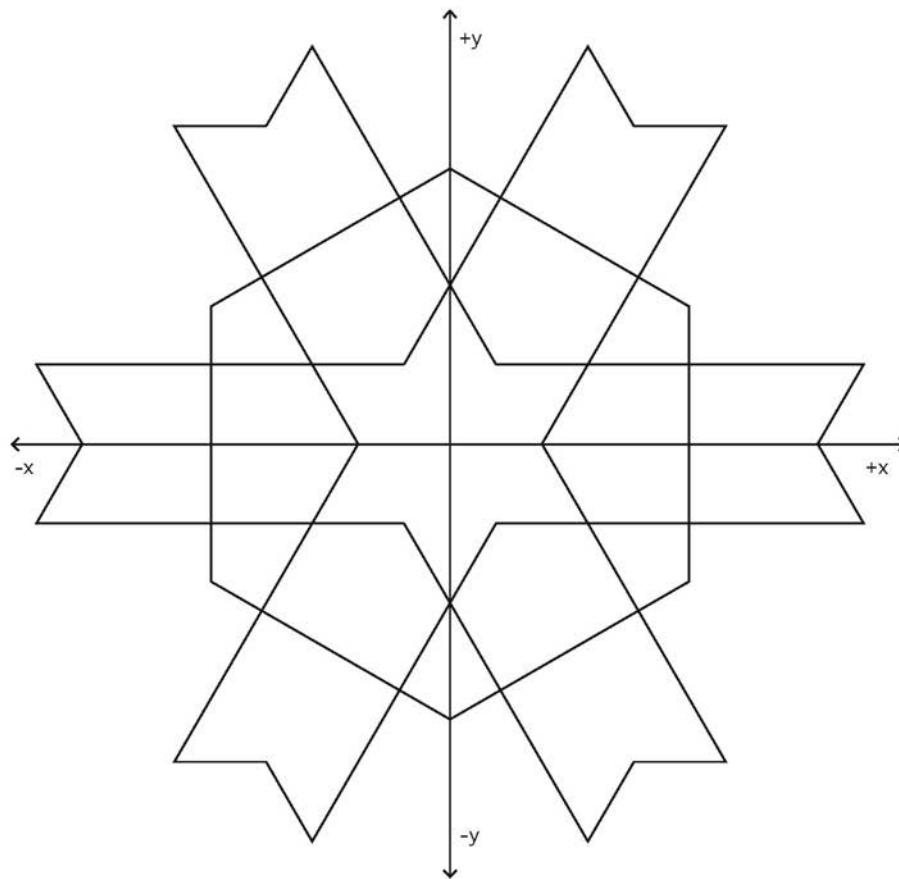
Aşağıdaki deseni inceleyin ve bu deseni oluşturan temel görsel bileşeni bulmaya çalışın



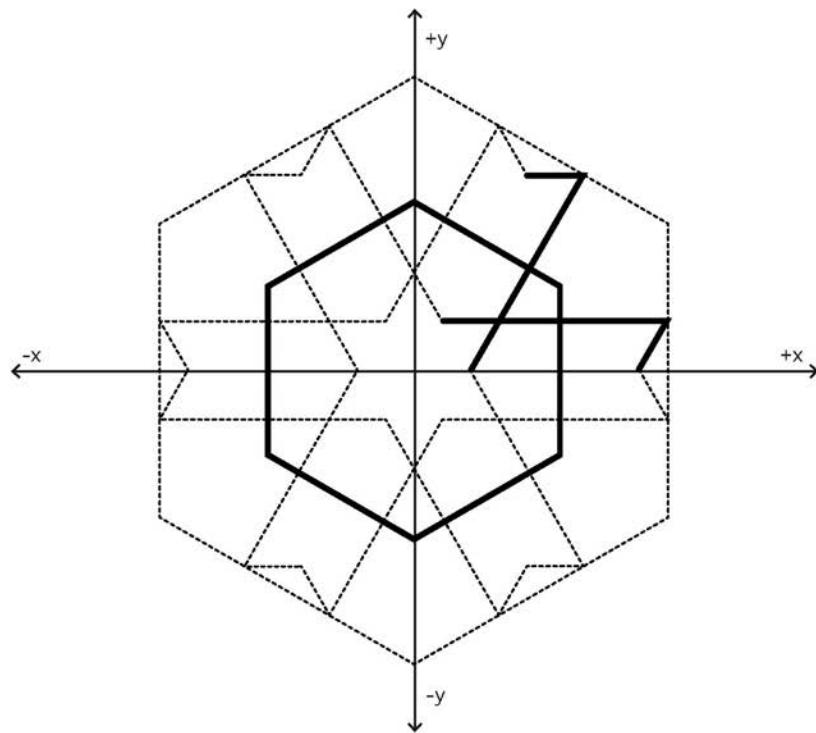
Motif

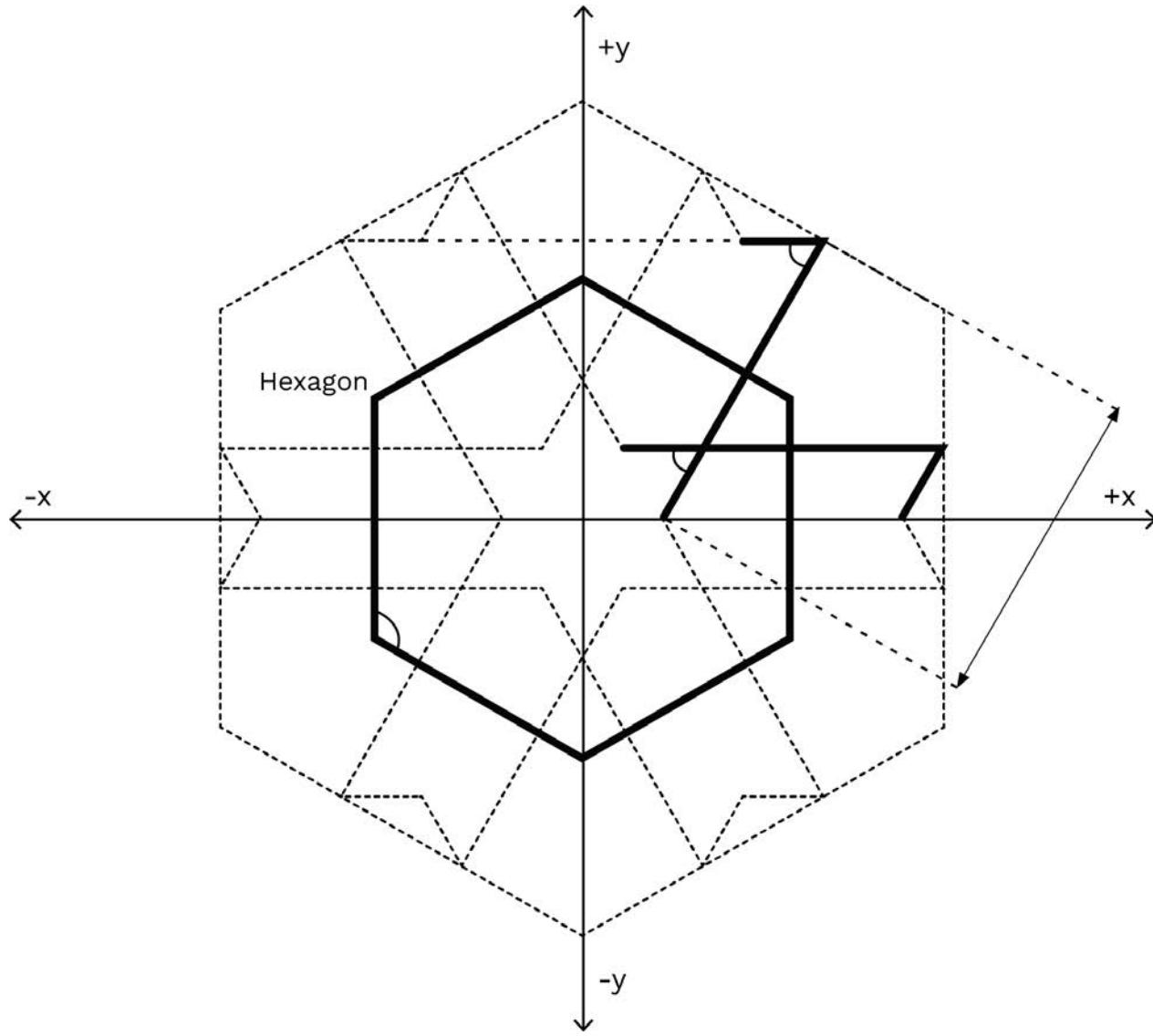


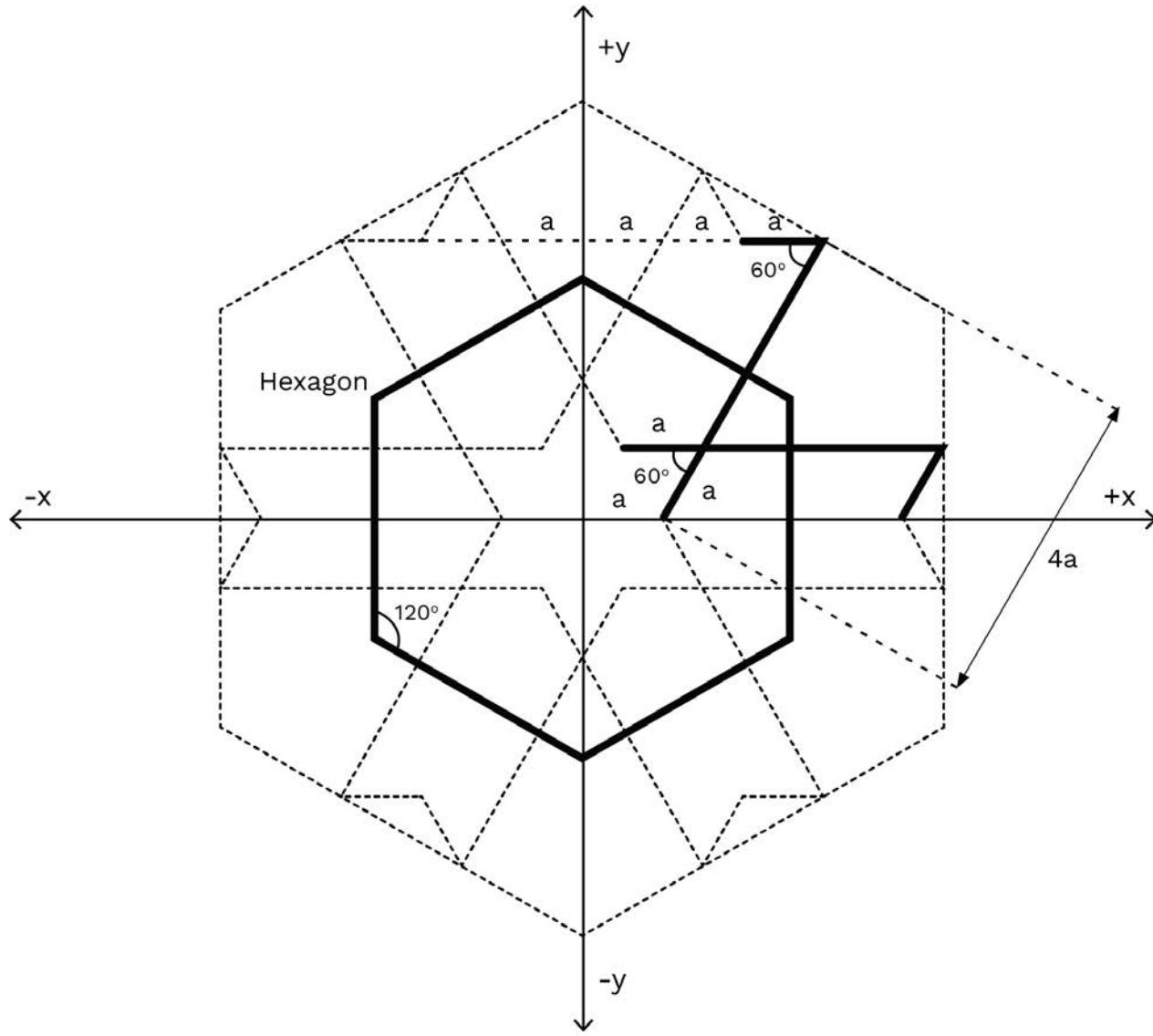
Motif



Temel Görsel Bileşini İnceleyelim

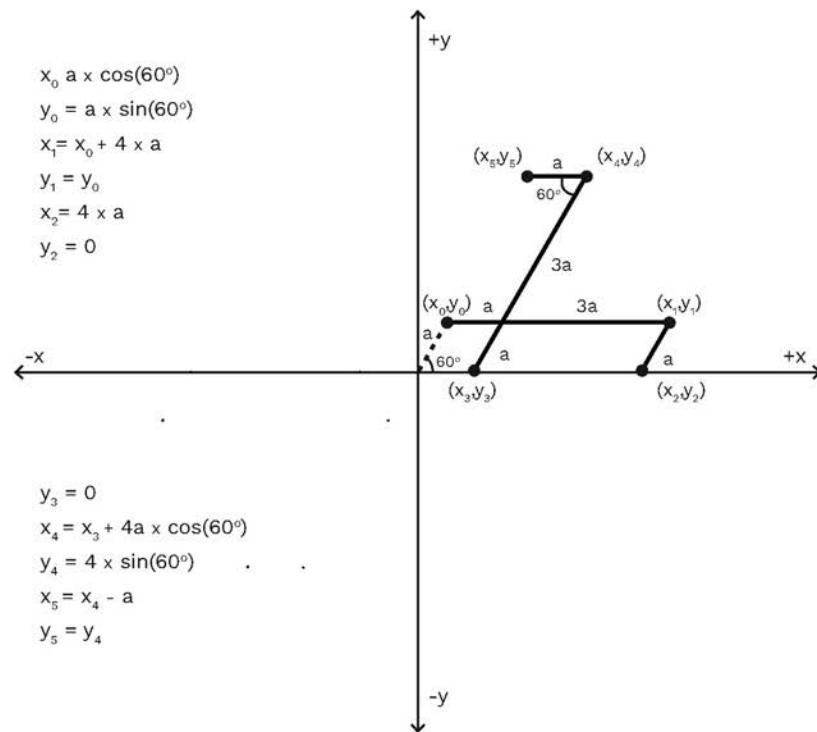






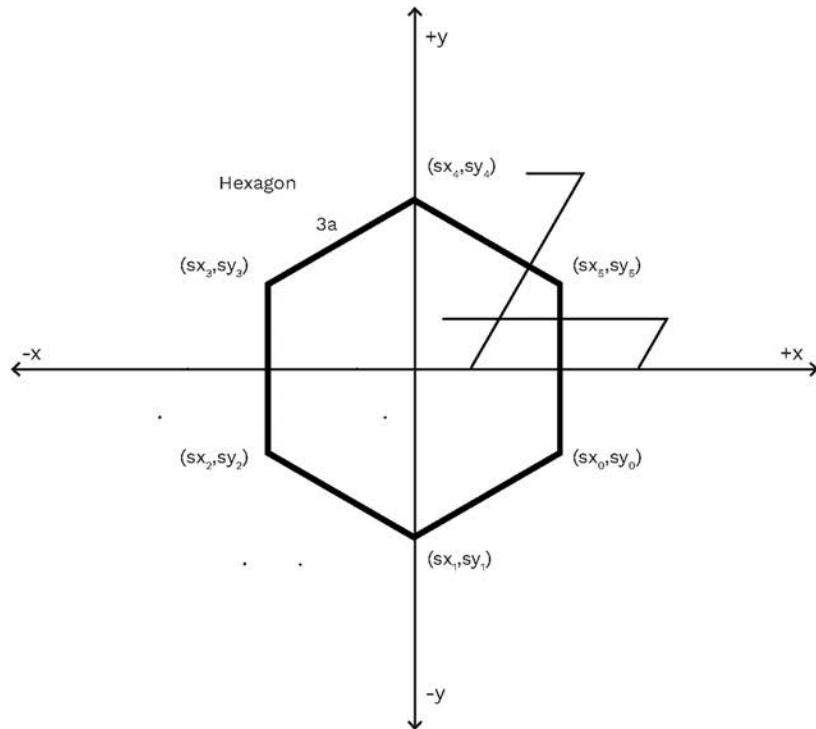
## Açıları ve Vertex noktalarını tespit etmek

Aşama 2: İlk olarak aşağıdaki şekli oluşturacağız.



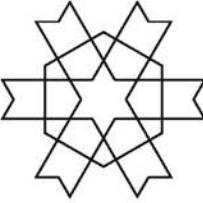
Açıları ve Vertex noktalarını tespit etmek

Aşama 3 : Şimdi dikkatimizi altigene verelim



## Motifi Oluşturmak

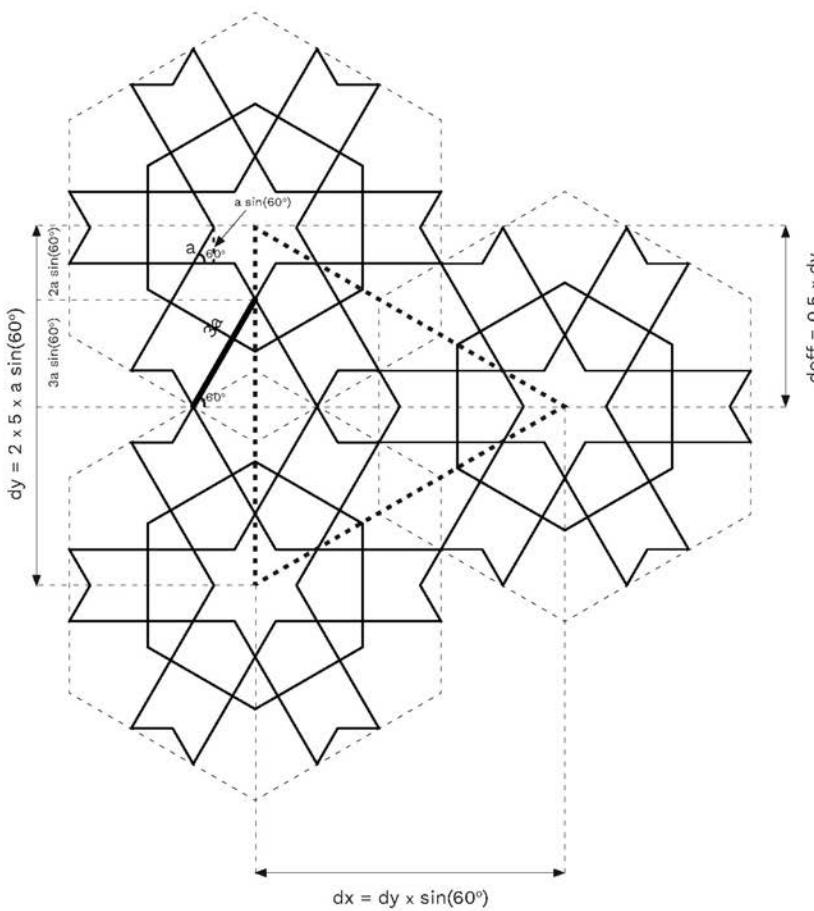
```
//scale factor  
let a = 40;  
  
function setup() {  
  createCanvas(400, 400);  
  angleMode(DEGREES);  
  noFill();  
  noLoop();  
}  
  
function draw() {  
  let x0, y0, x1, y1, x2, y2, x3, y3, x4, y4, x5, y5;  
  push();  
  translate(width * 0.5, height * 0.5);  
  //vertex points  
  x0 = a * cos(60);  
  y0 = -a * sin(60);  
  x1 = x0 + 4 * a;  
  y1 = y0;  
  x2 = 4 * a;  
  y2 = 0;  
  x3 = a;  
  y3 = 0;  
  x4 = x3 + 4 * a * cos(60);  
  y4 = -4 * a * sin(60);  
  x5 = x4 - a;  
  y5 = y4;  
  for (let i = 0; i < 6; i++) {  
    push();  
    rotate(60 * i);  
    beginShape();  
    vertex(x0, y0);  
    vertex(x1, y1);  
    vertex(x2, y2);  
    endShape();  
    beginShape();  
    vertex(x3, y3);  
    vertex(x4, y4);  
    vertex(x5, y5);  
    endShape();  
    pop();  
  }  
  pop();  
}
```



```
//hexagon  
//needs to be rotated 90 degrees to adjust the orientation  
rotate(90);  
let angle = 360 / 6;  
beginShape();  
for (let ang = 0; ang < 360; ang += angle) {  
  let sx = cos(ang) * 3 * a;  
  let sy = sin(ang) * 3 * a;  
  vertex(sx, sy);  
}  
endShape(CLOSE);  
pop();  
}
```

## Bezeme Yapısını İnceleyelim

Aşama 5 : Yerleştirmedeki  $dx$ ,  $dy$  ve doff değerlerini hesaplamamız gerekiyor.



## Bezeme Kodu

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

    display() {
        let x0, y0, x1, y1, x2, y2, x3, y3, x4, y4, x5, y5;
        x0 = a * cos(60);
        y0 = -a * sin(60);
        x1 = x0 + 4 * a;
        y1 = y0;
        x2 = 4 * a;
        y2 = 0;
        x3 = a;
        y3 = 0;
        x4 = x3 + 4 * a * cos(60);
        y4 = -4 * a * sin(60);
        x5 = x4 - a;
        y5 = y4;
        for (let i = 0; i < 6; i++) {
            push();
            rotate(60 * i);
            beginShape();
            vertex(x0, y0);
            vertex(x1, y1);
            vertex(x2, y2);
            endShape();
            beginShape();
            vertex(x3, y3);
            vertex(x4, y4);
            vertex(x5, y5);
            endShape();
            pop();
        }
    }

    //hexagon
    //needs to be rotated 90 degrees to adjust the orientation
    rotate(90);
    let angle = 360 / 6;
    beginShape();
    for (let ang = 0; ang < 360; ang += angle) {
        let sx = cos(ang) * 3 * a;
        let sy = sin(ang) * 3 * a;
        vertex(sx, sy);
    }
}

}
endShape(CLOSE);
}

//scale factor
let a = 16;
let motif = new Motif(a);
let nRow;
let nCol;
let dx, dy, doff;

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

    //even columns 0,2,4,....
    //odd columns 1,3,5,....
    dy = 2 * 5 * a * sin(60);
    dx = dy * sin(60);
    doff = dy * 0.5;

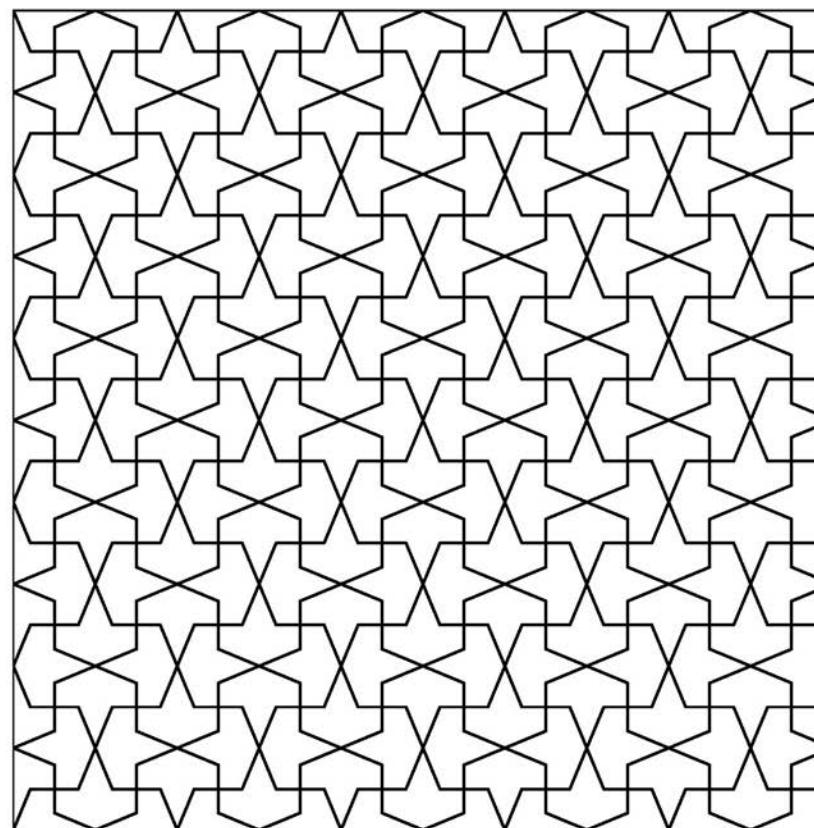
    //approximate the nRow and nCol values
    nRow = 1 + ceil(height / dy);
    nCol = 1 + ceil(width / dx);
}

function draw() {
    for (let c = 0; c < nCol; c++) {
        for (let r = 0; r < nRow; r++) {
            push();
            if (c % 2 == 1) {
                //columns 1,3,5,7
                translate(0, doff);
            }
            translate(dx * c, dy * r);
            motif.display();
            pop();
        }
    }
}
```

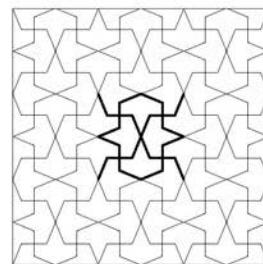
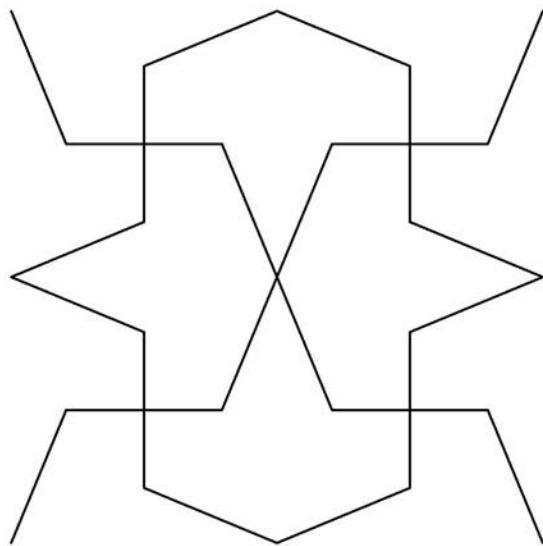
### **Geometrik Deseni Kodlamak**

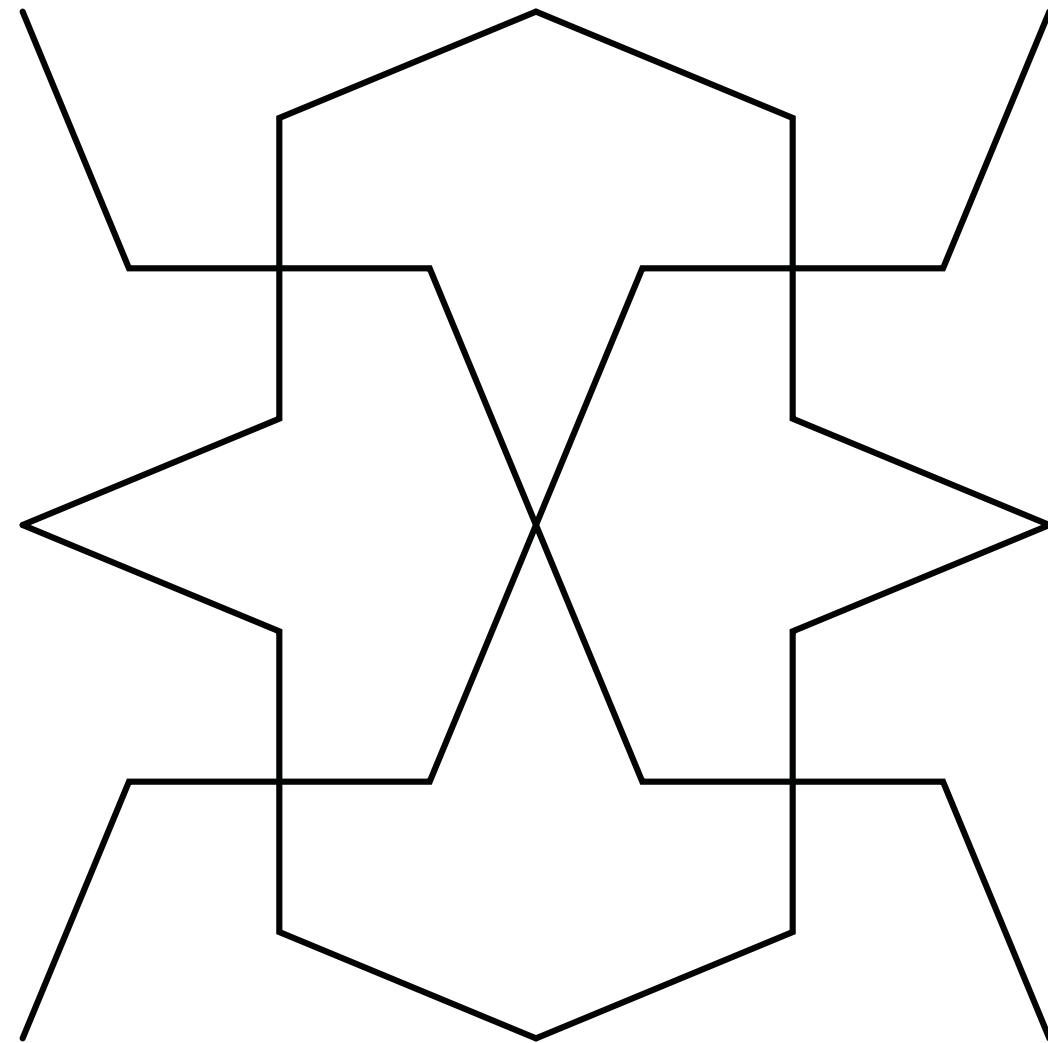
#### **Örnek 5**

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

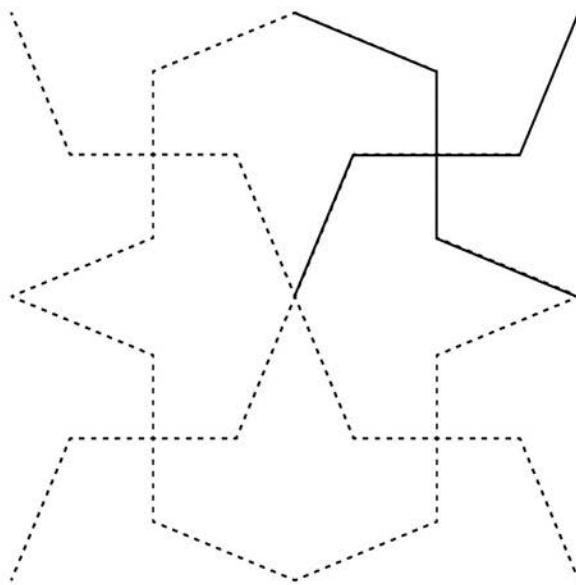


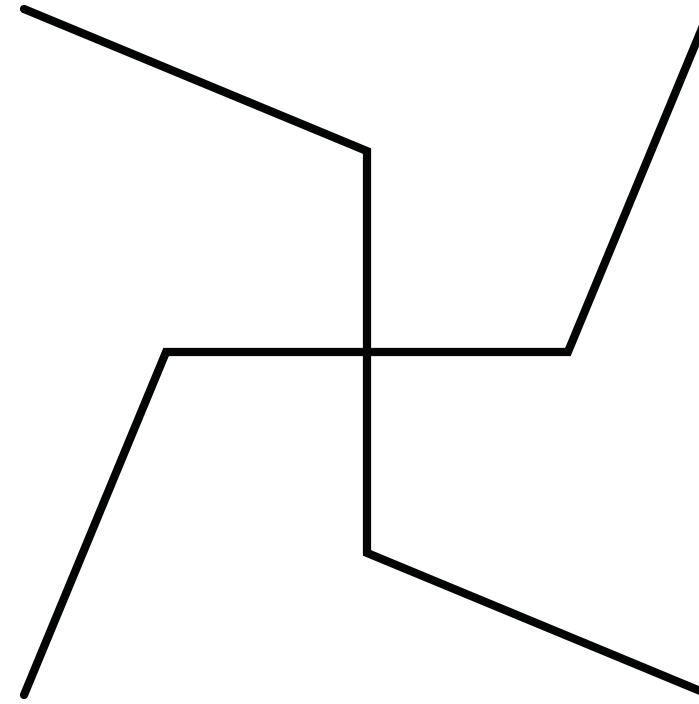
Motif

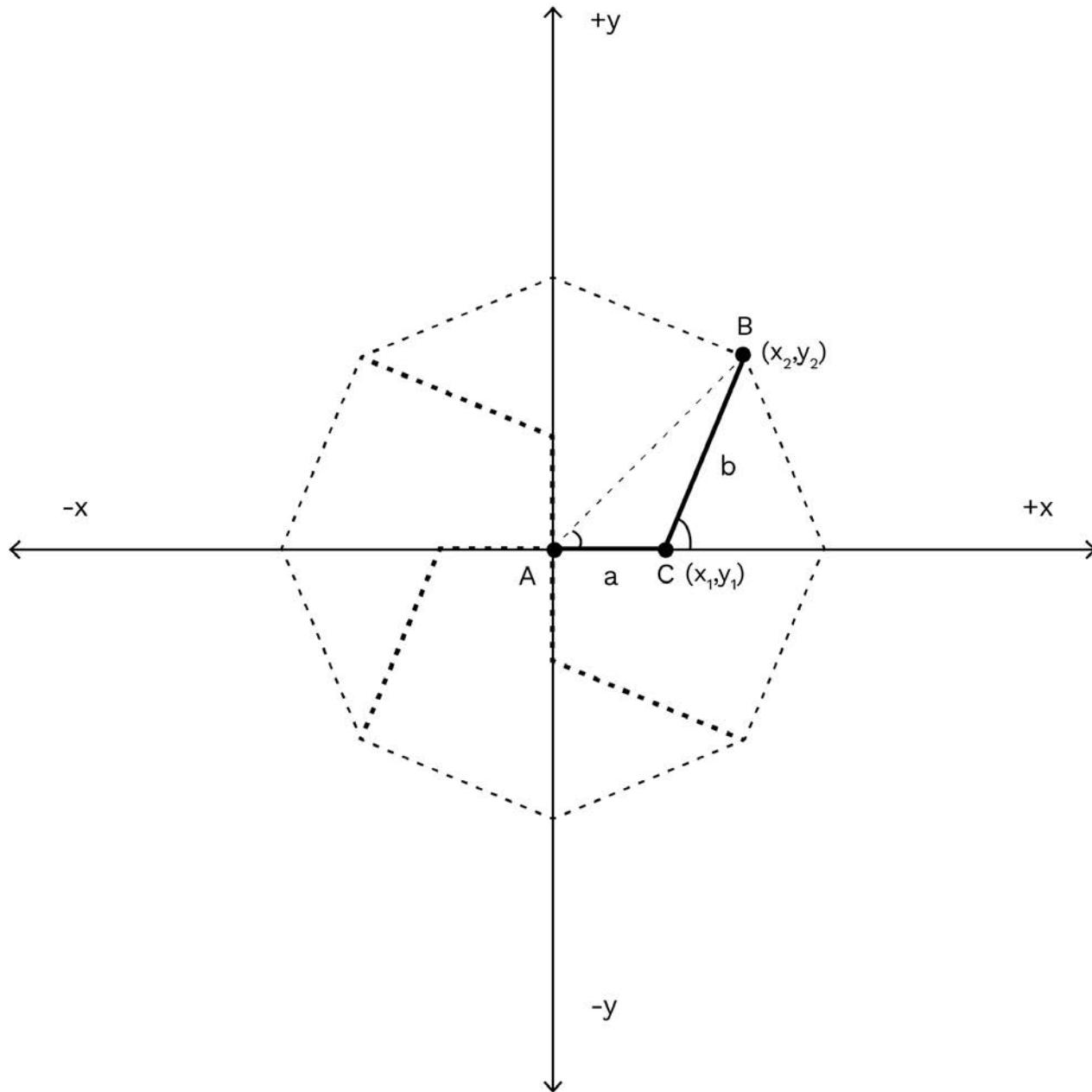




## Temel Görsel Bileşini İnceleyelim

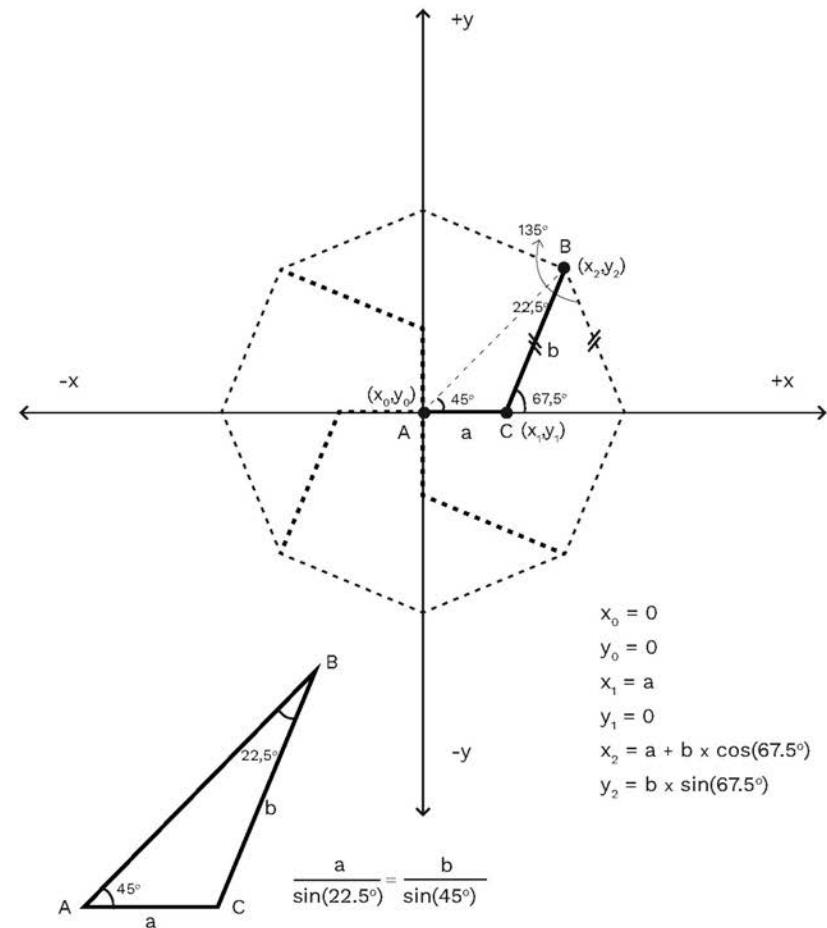






## Açıları ve Vertex noktalarını tespit etmek

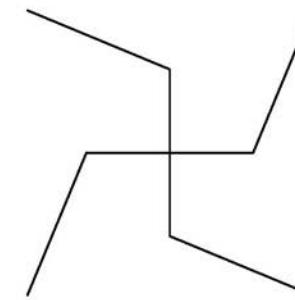
Aşama 1 : Vertex noktalarını bulalım



## 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();
}
```





Tokyo, Japonya



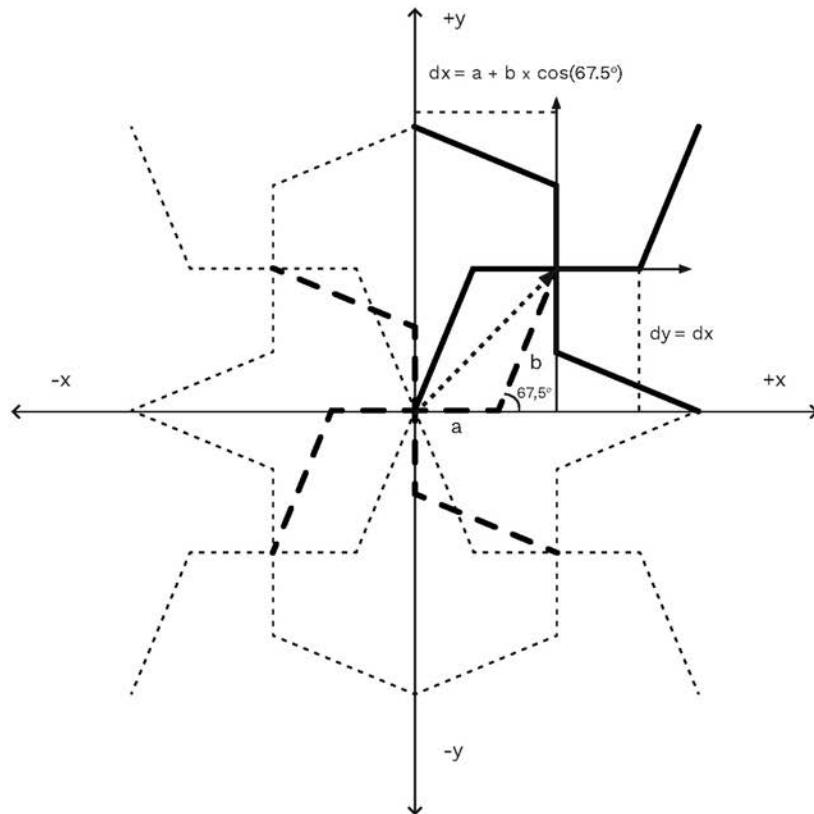


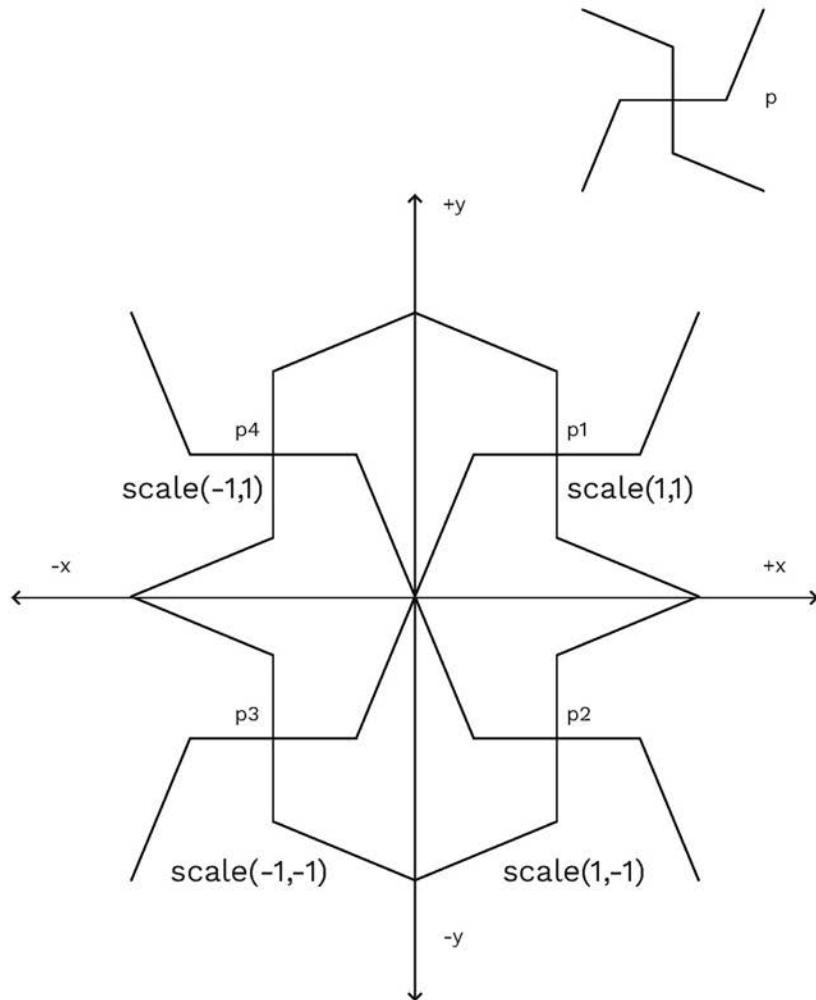
Efes Antik Kenti, Selçuk



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

yansımlar 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));
}

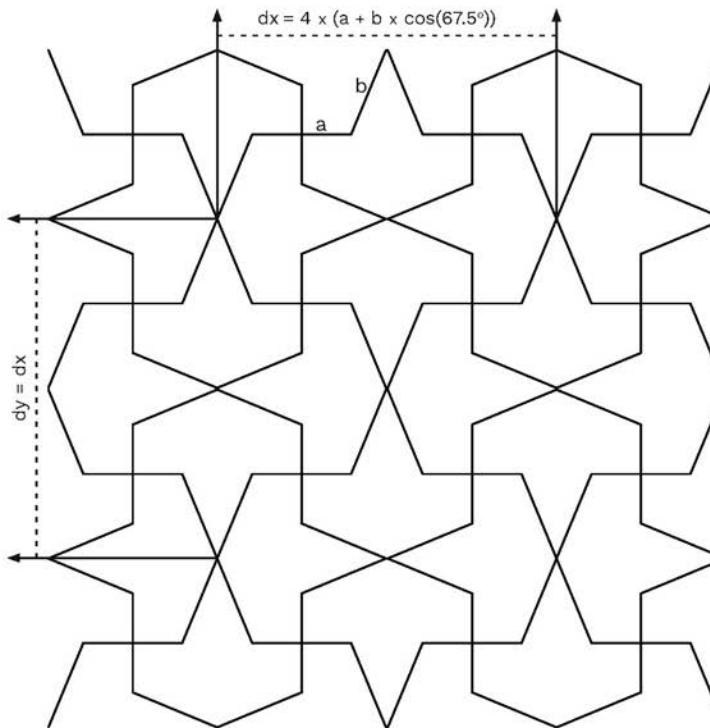
function draw() {
  noFill();
  let x0, y0, x1, y1, x2, y2;
  let dx = a + b * cos(67.5);
  let dy = dx;
  push();
  translate(width * 0.5, height * 0.5);
  for (let k = 0; k < 4; k++) {
    push();
    //mirroring
    switch (k) {
      case 0:
        scale(1, 1);
        break;
      case 1:
    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

## Motifi Oluşturmak

```
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();
    }
}
```

Teşekkürler.  
İletişimde kalalım!

Facebook/selcuk.artut

Instagram: selcukartut

Web: www.selcukartut.com

Email: selcukartut@gmail.com

