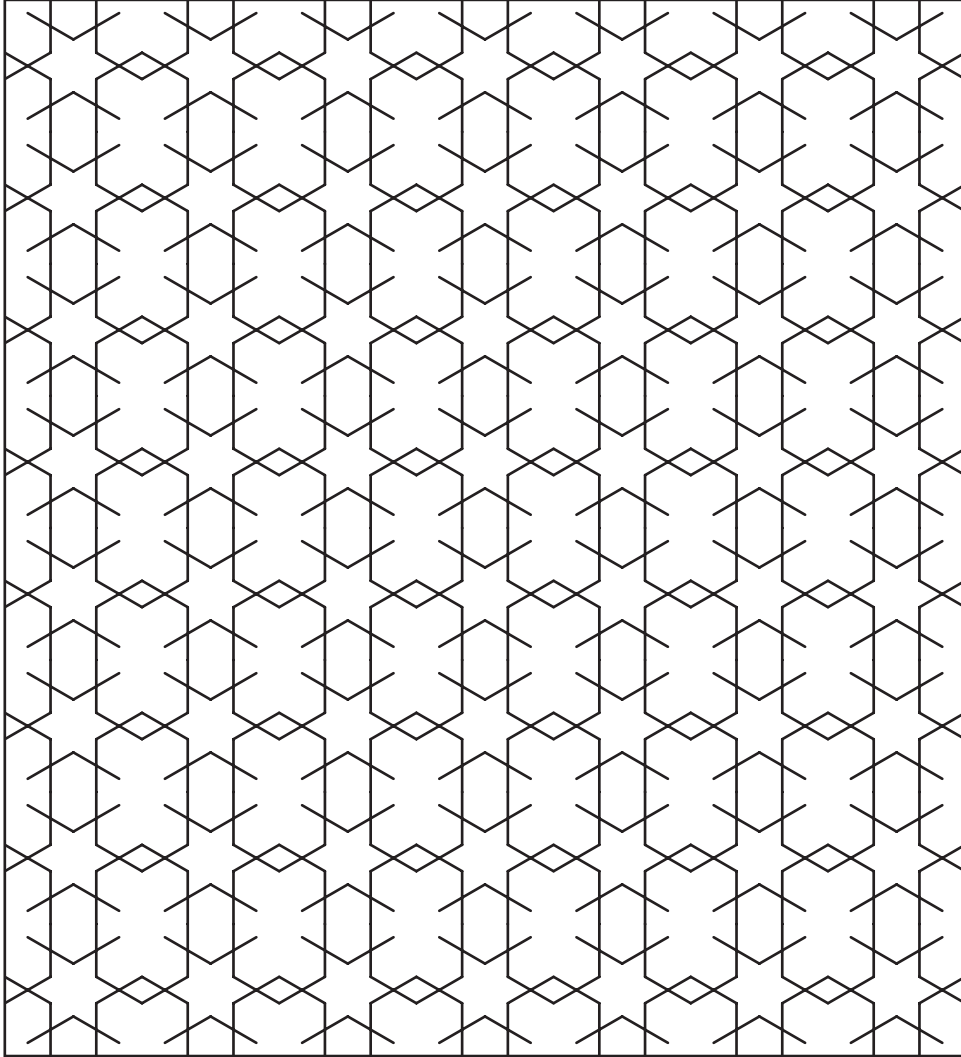


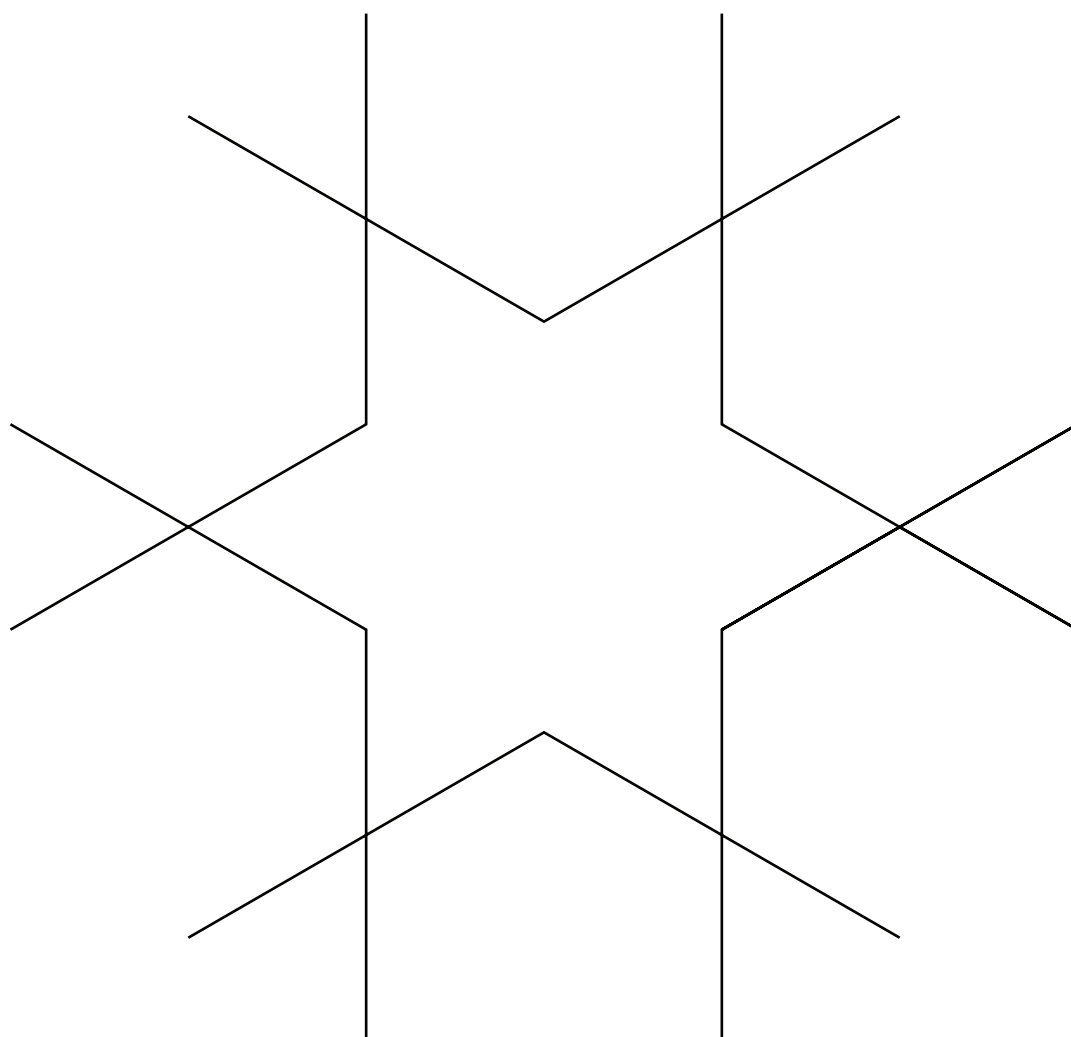
## Geometrik Deseni Kodlamak

### Örnek 1: Eşrefoğlu Cami, Beyşehir

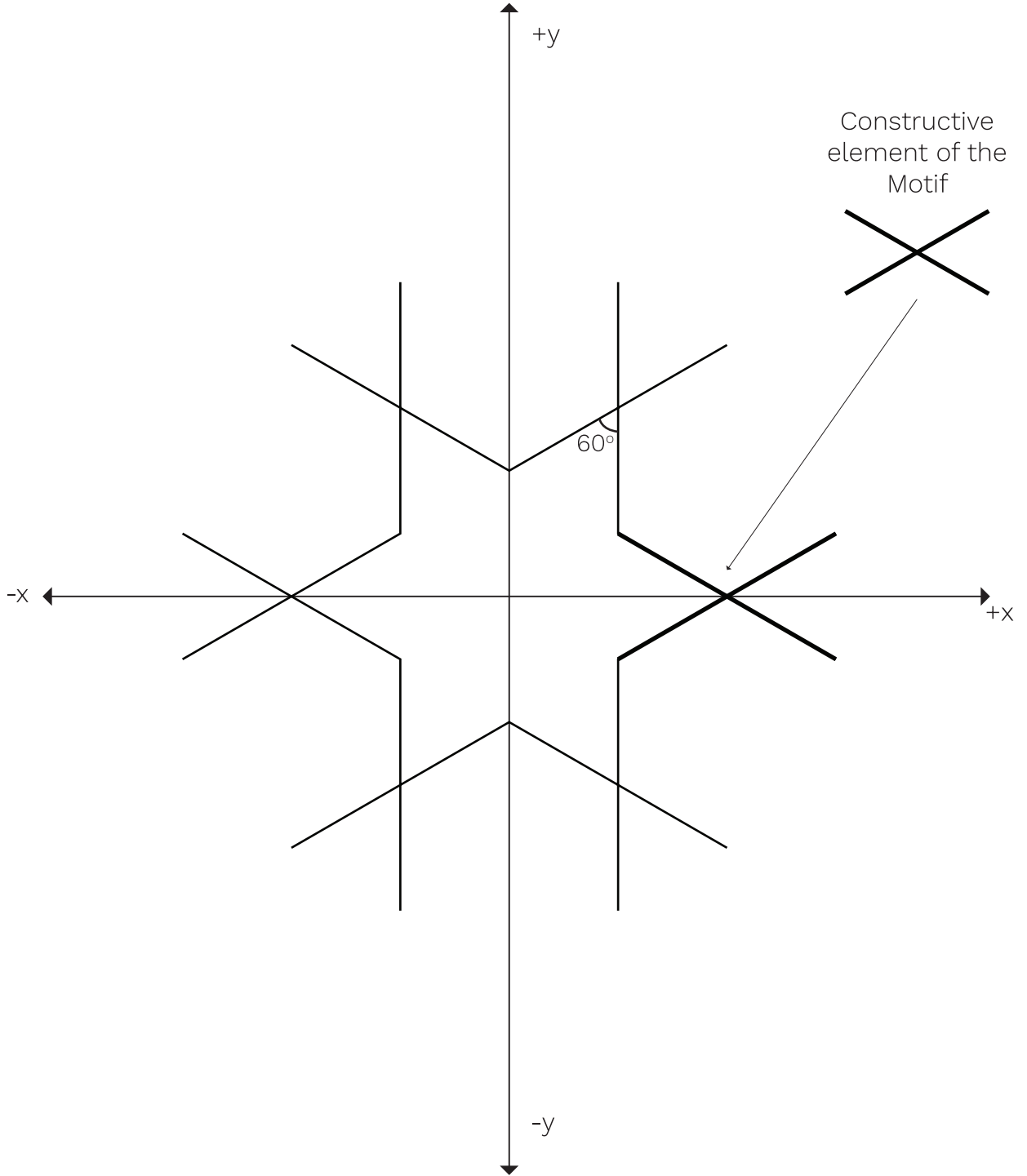
Aşağıdaki deseni inceleyin ve bu deseni oluşturan temel görsel bileşeni bulmaya çalışın.



## Motif



## Temel Görsel Bileşini İnceleyelim



# Açıları ve Vertex noktalarını tespit etmek

Aşama 1 : Vertex noktalarını bulalım

$$x_1 = a \times \cos(30^\circ)$$

$$y_1 = a \times \sin(30^\circ)$$

$$x_2 = -x_1$$

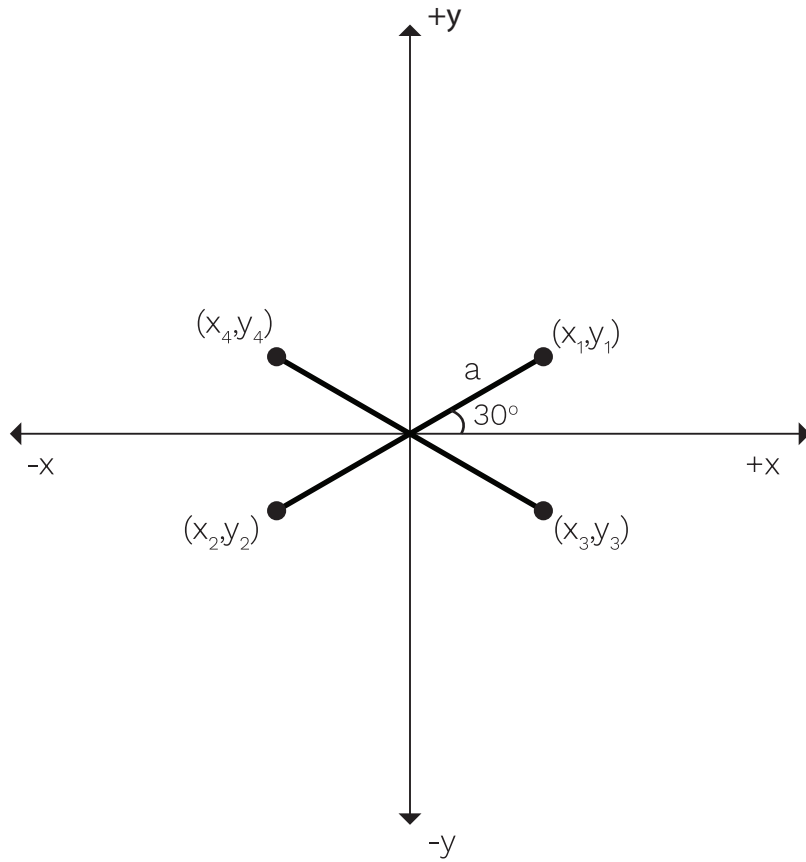
$$y_2 = -y_1$$

$$x_3 = x_1$$

$$y_3 = y_2$$

$$x_4 = x_2$$

$$y_4 = y_1$$



# Motifi Oluşturmak

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

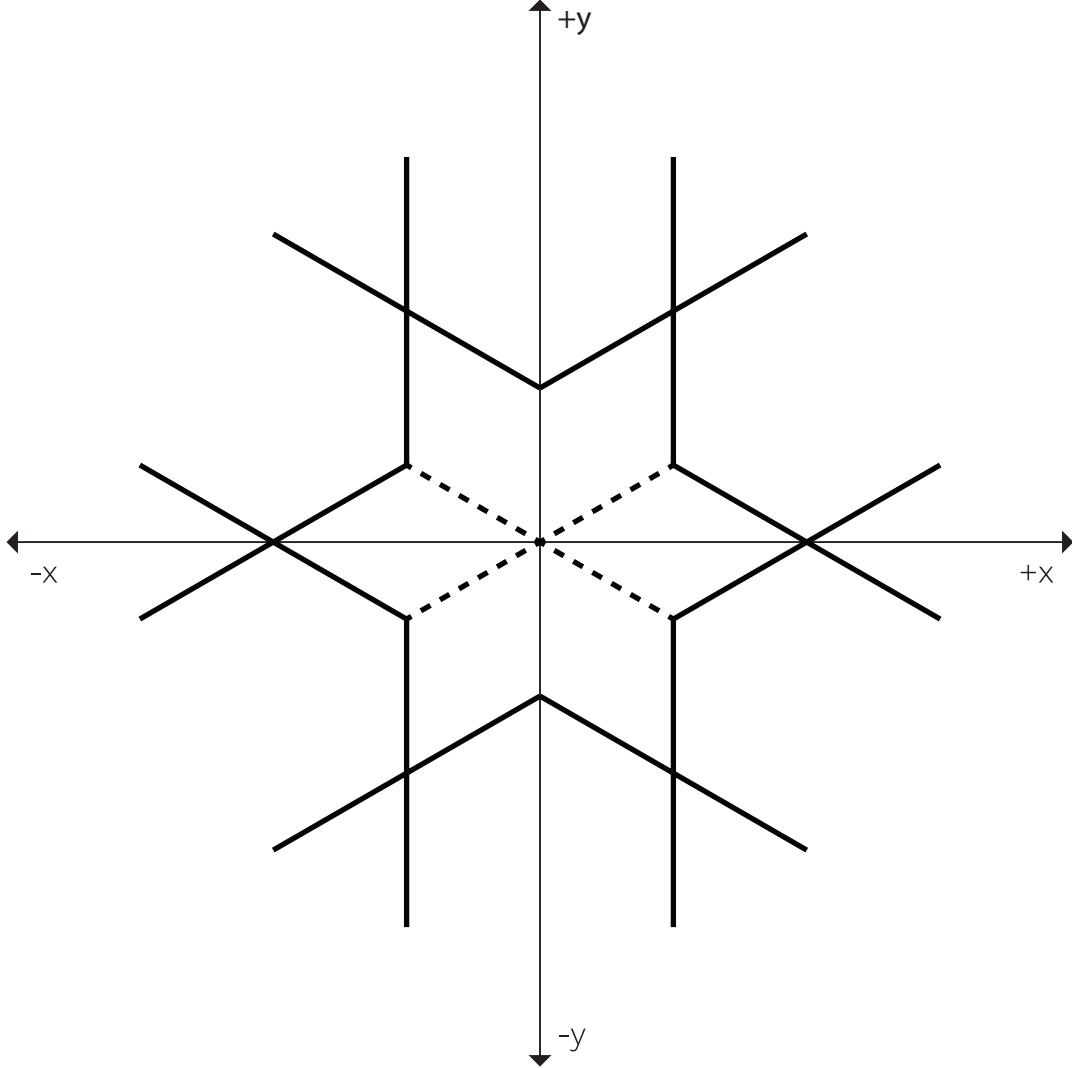
```
let a = 30;
```

```
function setup() {  
  createCanvas(600, 600);  
  angleMode(DEGREES);  
}  
function draw() {  
  let x1,y1,x2,y2,x3,y3,x4,y4;  
  background(255);  
  push();  
    translate(width*0.5, height*0.5);  
    //line one  
    beginShape();  
    x1 = a * cos(30);  
    y1 = a * sin(30);  
    vertex(x1,y1);  
    x2 = -1 * x1;  
    y2 = -1 * y1;  
    vertex(x2,y2);  
    endShape();  
    //line two  
    beginShape();  
    x3 = x1;  
    y3 = y2;  
    vertex(x3,y3);  
    x4 = x2;  
    y4 = y1;  
    vertex(x4,y4);  
    endShape();  
  pop();  
}
```

## Motifi Oluřturmak

Ařama 3 : Motifi, Temel Grsel Bileřeni kullanarak transformasyon fonksiyonları yardımı ile oluřturalım.

Algoritma: Temel Grsel Bileřeni yarı geniřlięinde saęa kaydır. Merkez etrafında altı defa dndr.



# Motifi Oluşturmak

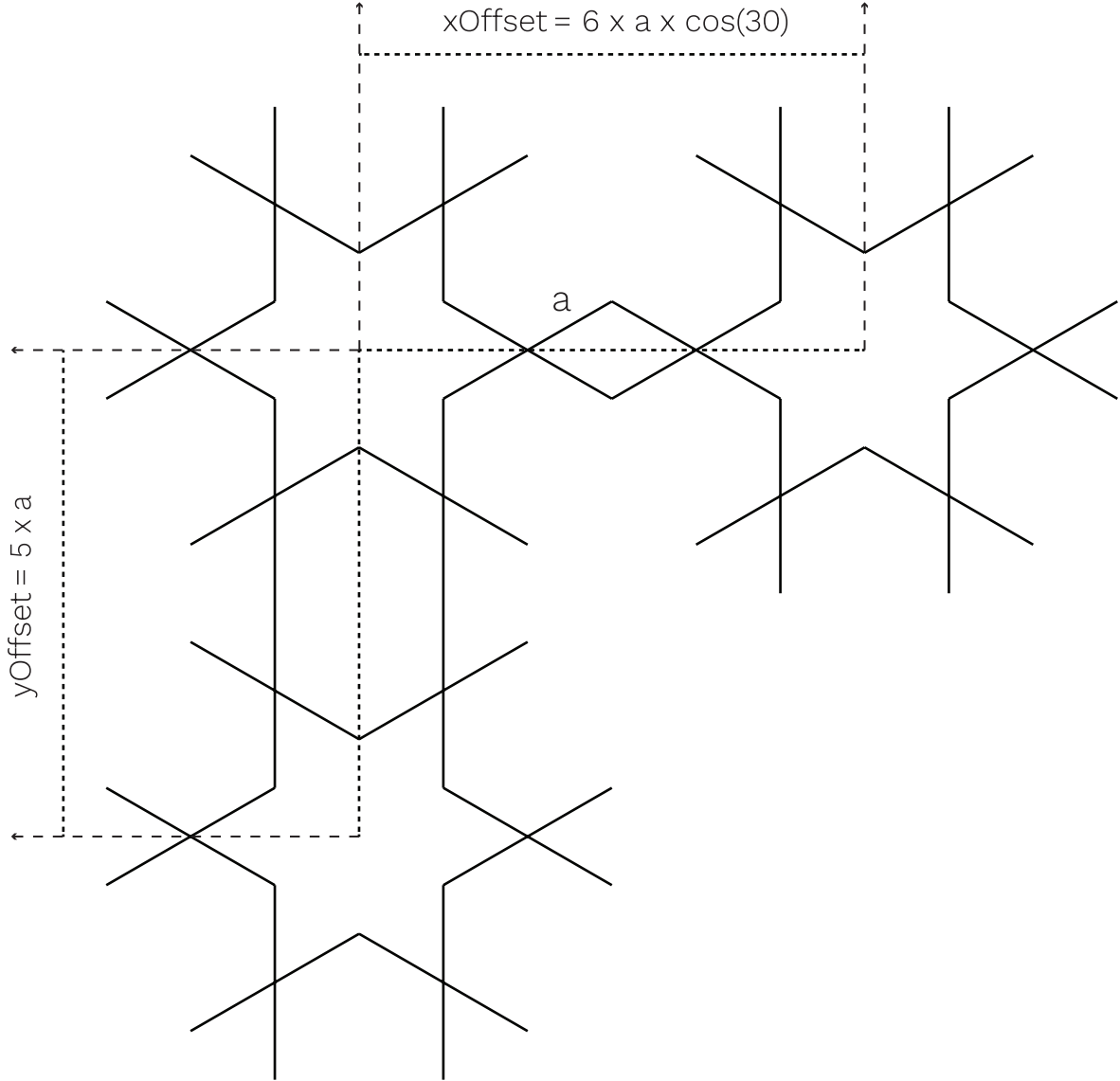
Aşama 4 : Motifi kodla oluşturmak için 6 tekrarlı bir loop döngüsü kullanacağız. Unutmayın transformasyon fonksiyonlarının sırası önemli!

```
let a = 30;

function setup() {
  createCanvas(400, 400);
  angleMode(DEGREES);
}
function draw() {
  let x1,y1,x2,y2,x3,y3,x4,y4;
  background(255);
  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();
        x1 = a * cos(30);
        y1 = a * sin(30);
        vertex(x1,y1);
        x2 = -1 * x1;
        y2 = -1 * y1;
        vertex(x2,y2);
        endShape();
        //line two
        beginShape();
        x3 = x1;
        y3 = y2;
        vertex(x3,y3);
        x4 = x2;
        y4 = y1;
        vertex(x4,y4);
        endShape();
      pop();
    }
  pop();
}
```

## Bezeme Yapısını İnceleyelim

Aşama 5 : Yukarı ve aşağıya kaymaları belirleyecek xoffset ve yoffset değerlerini hesaplayalım.





# Bezeme Kodu

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

  display() {
    let x1, y1, x2, y2, x3, y3, x4, y4;
    for (let i = 0; i < 6; i++) {
      push();
      rotate(i * 60);
      translate(cos(30) * this.a * 2, 0);
      //line one
      beginShape();
      x1 = this.a * cos(30);
      y1 = this.a * sin(30);
      vertex(x1, y1);
      x2 = -1 * x1;
      y2 = -1 * y1;
      vertex(x2, y2);
      endShape();
      //line two
      beginShape();
      x3 = x1;
      y3 = y2;
      vertex(x3, y3);
      x4 = x2;
      y4 = y1;
      vertex(x4, y4);
      endShape();
      pop();
    }
  }
}
```

```

let a = 16;
let motives = []; // Declare array
let nRow;
let nCol;
let xoffset, yoffset;

function setup() {
  createCanvas(600, 600);
  angleMode(DEGREES);

  xoffset = a * cos(30) * 6;
  yoffset = 5 * a;

  //approximate the nRow and nCol values
  //try and see the results
  nRow = ceil(width / xoffset);
  nCol = ceil(height / yoffset);

  //generate the motives array
  for (let i = 0; i < nRow * nCol; i++) {
    motives.push(new Motif(a));
  }
}

function draw() {
  background(255);
  noFill();
  noLoop();
  for (let c = 0; c < nCol; c++) {
    for (let r = 0; r < nRow; r++) {
      push();
      translate(xoffset * c, yoffset * r);
      motives[c + r * nCol].display();
      pop();
    }
  }
}

```