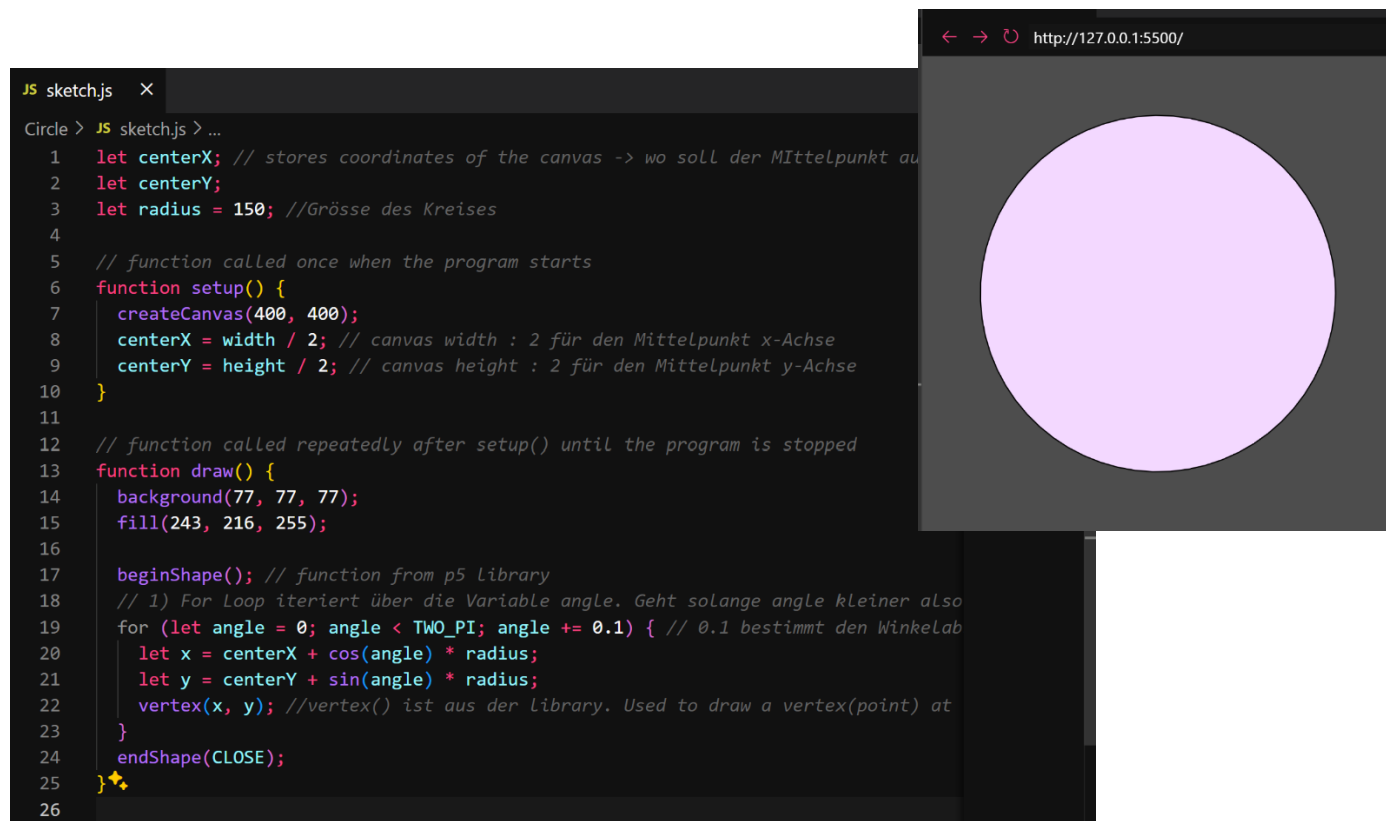


Algorithmic Art – Der Kreis

1) Kreis

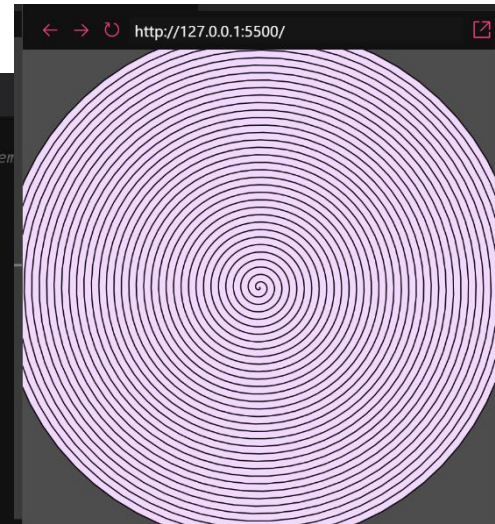
Logik hinter der Implementation: Via Polarkoordinaten, die einen Punkt beschreiben durch seinen Abstand zum Mittelpunkt und dem Winkel, den er zur x-Achse bildet. Dann mit Trigonometrie wird der Punkt auf dem Kreis berechnet.

Mit `endShape()` werden die Punkte dann verbunden.

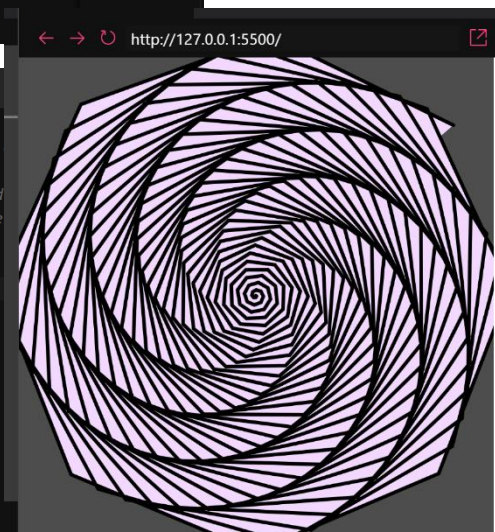


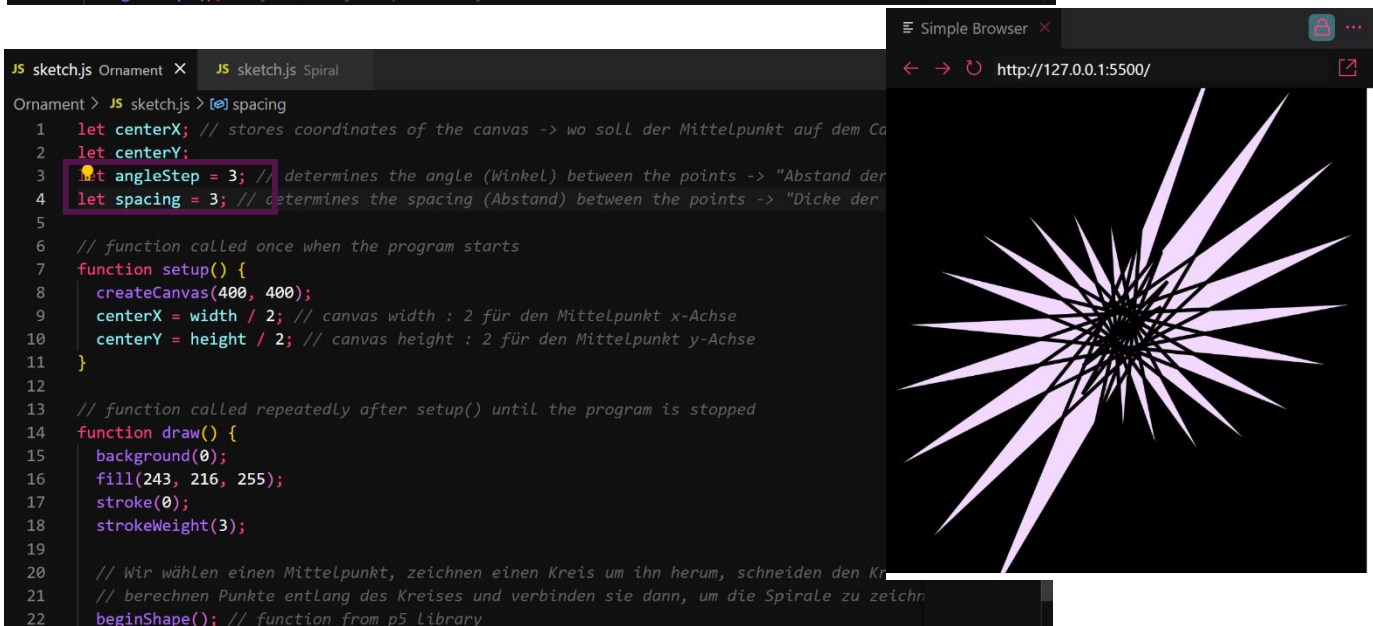
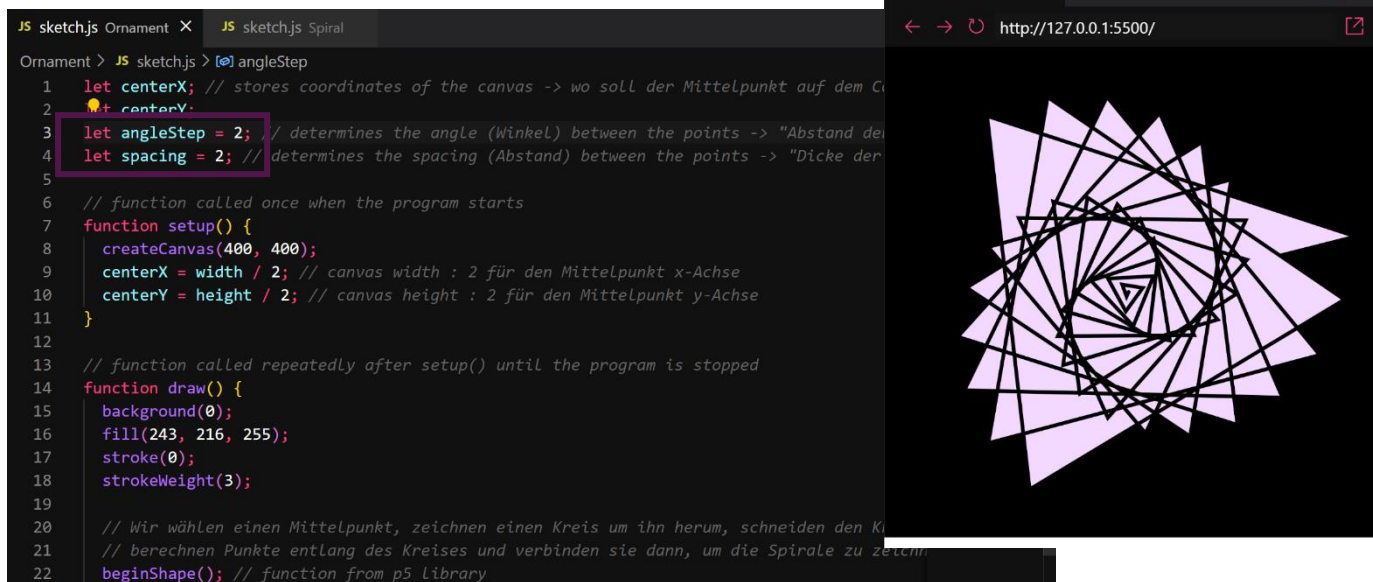
2) Spirale

```
JS sketchjs X
Spiral > JS sketchjs > draw
1 let centerX; // stores coordinates of the canvas -> wo soll der Mittelpunkt auf dem
2 let centerY;
3 let angleStep = 0.1; // determines the angle (Winkel) between the points
4 let spacing = 1; // determines the spacing (Abstand) between the points
5
6 // function called once when the program starts
7 function setup() {
8   createCanvas(400, 400);
9   centerX = width / 2; // canvas width : 2 für den Mittelpunkt x-Achse
10  centerY = height / 2; // canvas height : 2 für den Mittelpunkt y-Achse
11 }
12
13 // function called repeatedly after setup() until the program is stopped
14 function draw() {
15   background(77, 77, 77);
16   fill(243, 216, 255);
17
18   beginShape(); // function from p5 Library
19   for (let angle = 0; angle < 70 * PI; angle += angleStep) {
20     let r = angle * spacing;
21     let x = centerX + r * cos(angle);
22     let y = centerY + r * sin(angle);
23     vertex(x, y); //vertex() ist aus der Library. Used to draw a vertex(point) at the calc
24   }
25   endShape();
26
27   // angleStep += 0.01; //Berechnung des Winkels in der Loop immer wieder neu. Resultiert
28   // spacing += 0.01; // Berechnung des Abstands in der Loop immer wieder neu. Resultiert
29 }
30
31
```

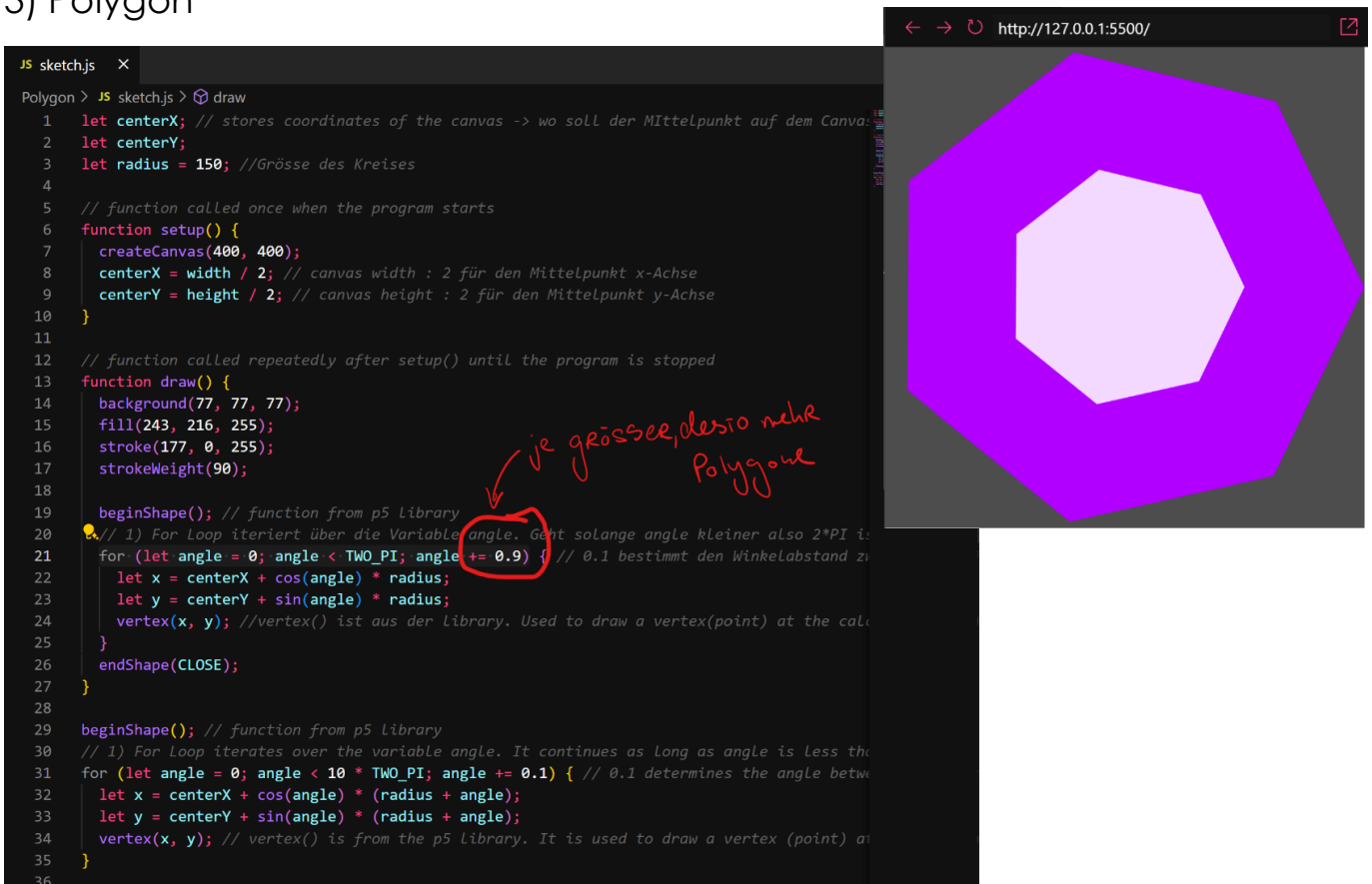


```
JS sketchjs X
Spiral > JS sketchjs > setup
1 let centerX; // stores coordinates of the canvas -> wo soll der Mittelpunkt auf dem
2 let centerY;
3 let angleStep = 0.8; // determines the angle (Winkel) between the points -> "Abstand
4 let spacing = 1; // determines the spacing (Abstand) between the points -> "Dicke de
5
6 // function called once when the program starts
7 function setup() {
8   createCanvas(400, 400);
9   centerX = width / 2; // canvas width : 2 für den Mittelpunkt x-Achse
10  centerY = height / 2; // canvas height : 2 für den Mittelpunkt y-Achse
11 }
12
13 // function called repeatedly after setup() until the program is stopped
14 function draw() {
15   background(77, 77, 77);
16   fill(243, 216, 255);
17   stroke(0);
18   strokeWeight(3);
19
20   // Wir wählen einen Mittelpunkt, zeichnen einen Kreis um ihn herum, schneiden den Kreis
21   // berechnen Punkte entlang des Kreises und verbinden sie dann, um die Spirale zu zeich
22   beginShape(); // function from p5 Library
23   for (let angle = 0; angle < 70 * PI; angle += angleStep) { //macht 70 Iterationen. PI er
24     let r = angle * spacing; // r wird immer größer, da angle immer größer wird. D.h. der
25     let x = centerX + r * cos(angle); //
26     let y = centerY + r * sin(angle);
27     vertex(x, y); //vertex() ist aus der Library. Used to draw a vertex(point) at the calc
28   }
29   endShape();
30
31   // angleStep += 0.01; //Berechnung des Winkels in der Loop immer wieder neu. Resultiert
32   // spacing += 0.01; // Berechnung des Abstands in der Loop immer wieder neu. Resultiert
33 }
34
35
```





3) Polygon



4) Ornament

```
JS sketch.js Ornament X JS sketch.js Circle
Ornament > JS sketch.js > ...
1 function setup(){
2   createCanvas(400, 400);
3   angleMode(DEGREES); // Winkel in Grad einstellen, wegen #persönlicher Präferenz
4   colorMode(HSB); // Es braucht Transparenz, damit die einzelnen Kreise nicht überlagern
5   background(225);
6 }
7 var c = 150;
8 var angle = 0; // c = Steuerung der Farbe, angle = Rotation der Zeichnung
9 function draw(){
10  stroke(255); //setzt die Farbe der Kontur auf Weiss
11  translate(width/2, height/2);
12  fill(c, 30, 30, 0.1); // (Farbe, Sättigung, Helligkeit, Transparenz)
13  // rotate(angle);
14
15  for(var i=0; i<12; i++){
16    ellipse(cos(30*i)*100, sin(30*i)*400, 400); // Grosse Kreise
17
18    push(); // Speichert den aktuellen Füllfarbenzustand der grossen Kreise. Braucht e
19    fill(255-c, 40, 80, 0.1);
20    ellipse(sin(30*i)*100, cos(30*i)*100, 200); // Kleine Kreise
21    pop(); // hier wird wieder der ursprüngliche Füllfarbenzustand (vor push) hergeste
22  }
23 }
24
25
```

