

# Intro til generativ kunst

Sondre Bakke

# Agenda

Nå - Kjapt om Bekk 

Snart - Ikke så kjapt om generativ kunst 

Snart+ - Det jobbes med generativ kunst 

18:30 - Mat fra McDonald's 

20:00 - Kursslutt og start av mingling med spill og drikke! 

# Hvem møter du i dag?



**Emilie Lia-Rognli**  
Informatikk  
reMarkable



**Sigurd Hynne**  
Datalogi  
Enova



**Sondre Bakke**  
I&IKT  
Utdanningsdirektoratet



**Therese Federl**  
Informatikk  
Statens Vegvesen



**Tor Martin Wang**  
Datalogi  
Domstoladministrasjonen

# Kjapt om Bekk

# Hva

Norsk konsulentselskap

Oslo og Trondheim



# Hvem

Tre avdelinger

Folk og Fag



# Hvordan

Vi jobber best sammen

Tverrfaglig



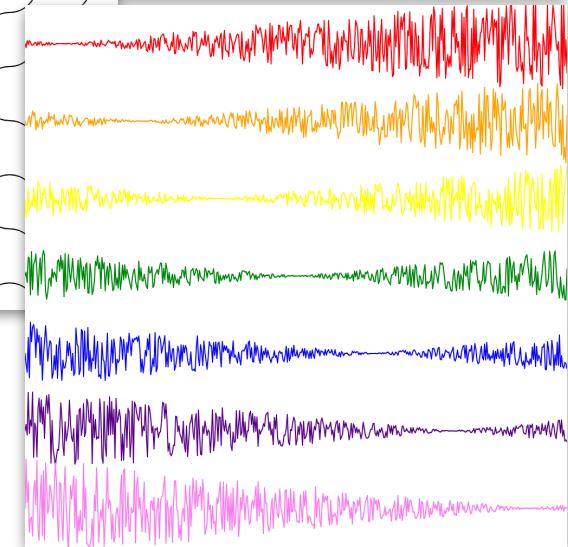
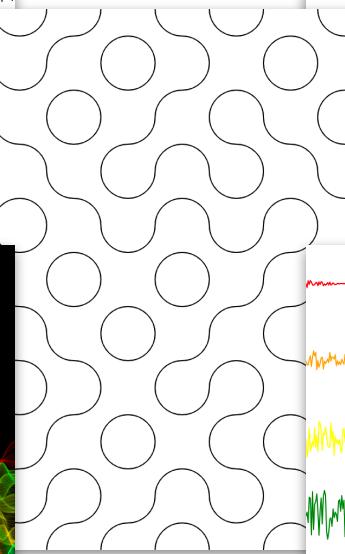
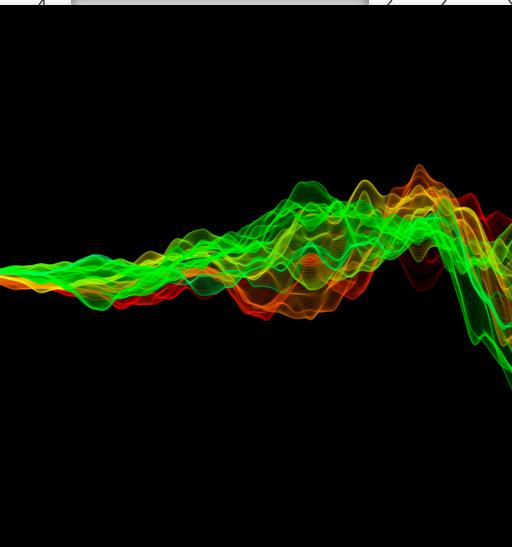
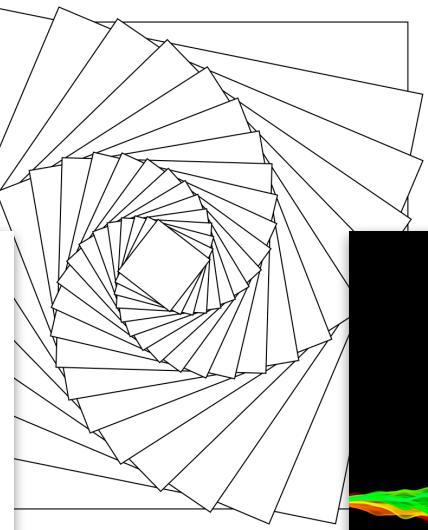
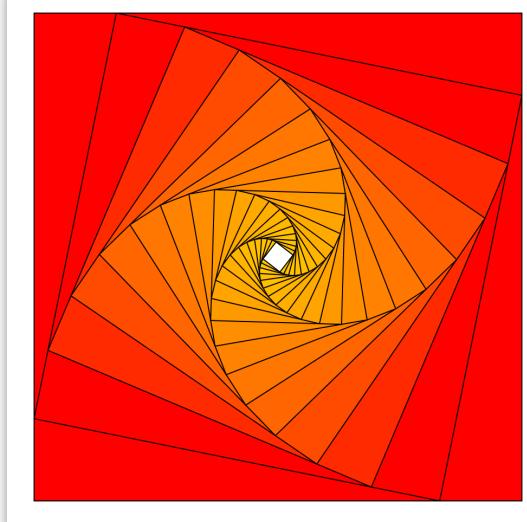
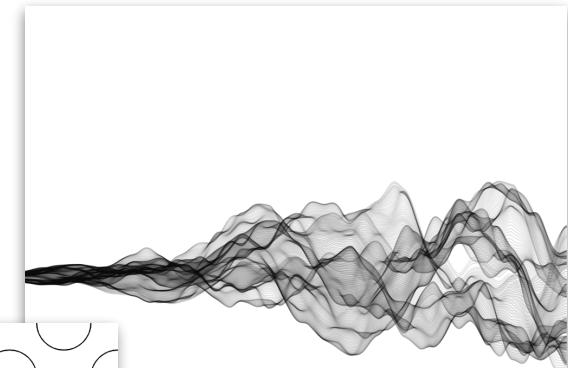
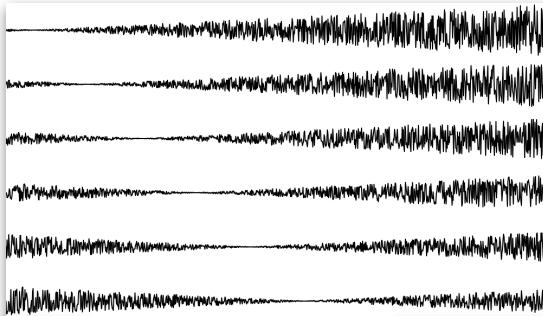
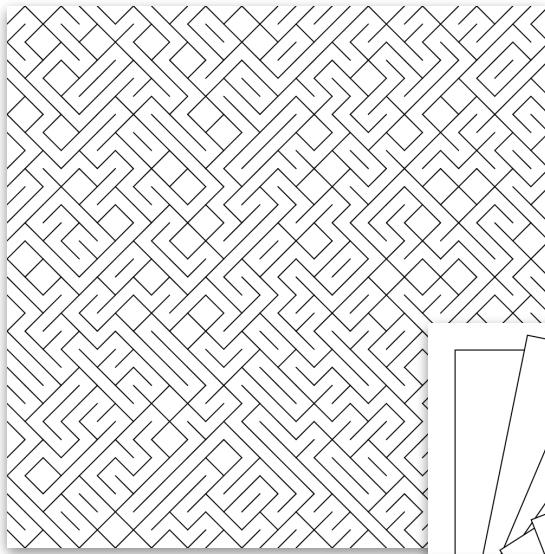
# Intro til generativ kunst

# Hva er generativ kunst?

“Generativ kunst er kunst som helt eller delvis er laget med hjelp av et autonomt system”

Algoritmisk kunst

# Hva er generativ kunst?



p5.js

# p5.js online editor

The screenshot shows the p5.js online editor interface. At the top, there is a navigation bar with the p5 logo, File, Edit, Sketch, and Help menus, and language selection for English. A red oval highlights the "Log in or Sign up" button. Below the navigation bar are control buttons for play/pause, auto-refresh, and grateful curtain, along with a settings gear icon.

The main area consists of two side-by-side panes. The left pane is labeled "sketch.js" and contains the following code:

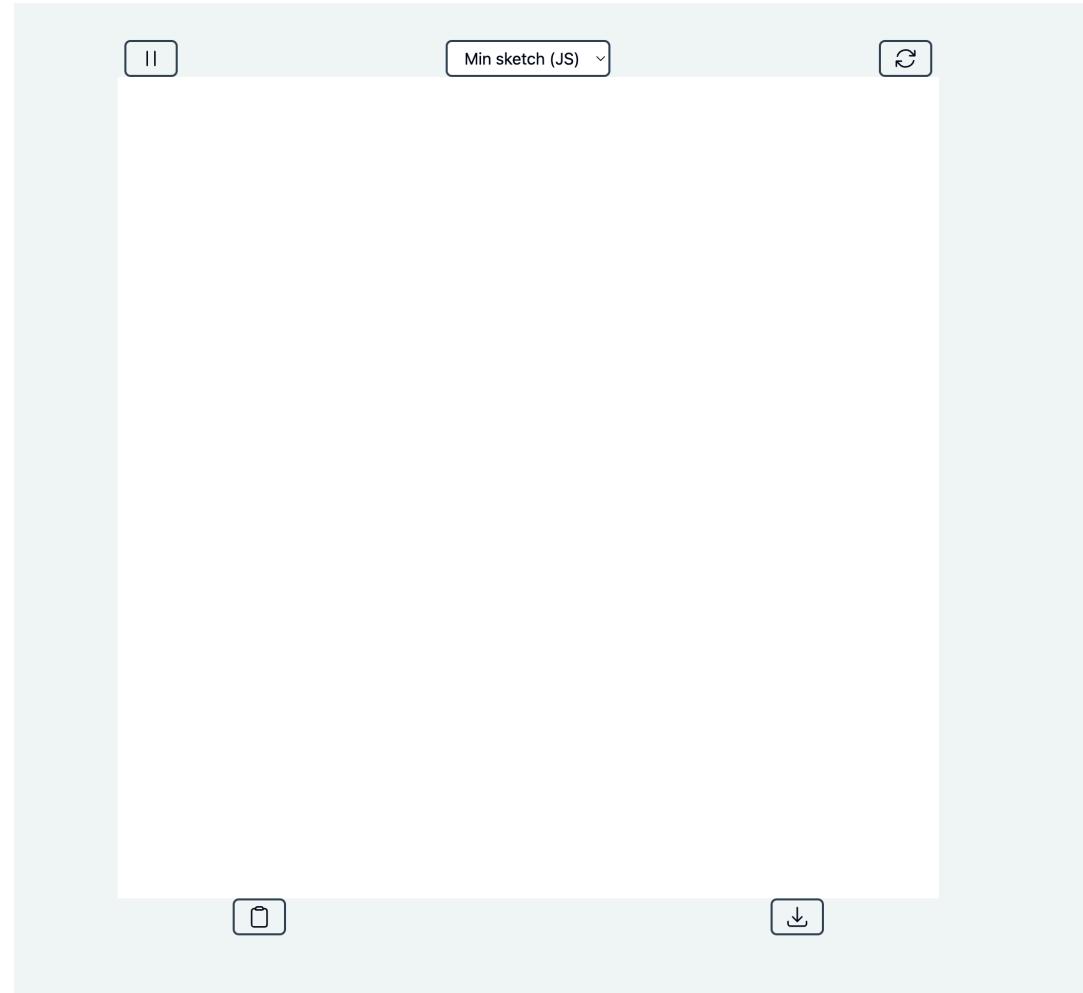
```
1 function setup() {
2   createCanvas(400, 400);
3 }
4
5 function draw() {
6   background(220);
7 }
```

The right pane is labeled "Preview" and shows a solid light gray canvas.

At the bottom, there is a "Console" tab with a "Clear" dropdown menu and a single character input field containing a greater-than sign (>).

# Sondres Superkule Github Repo™

- Styggere visning (subjektivt)
- Valg mellom TS og JS
- Filsystem
- Bytte mellom flere sketcher
- Knapper for å kopiere eller laste ned bilde



# Sondres Superkule Github Repo™

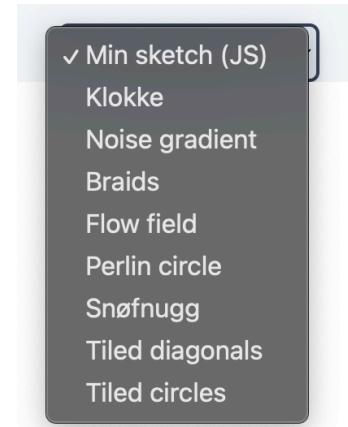
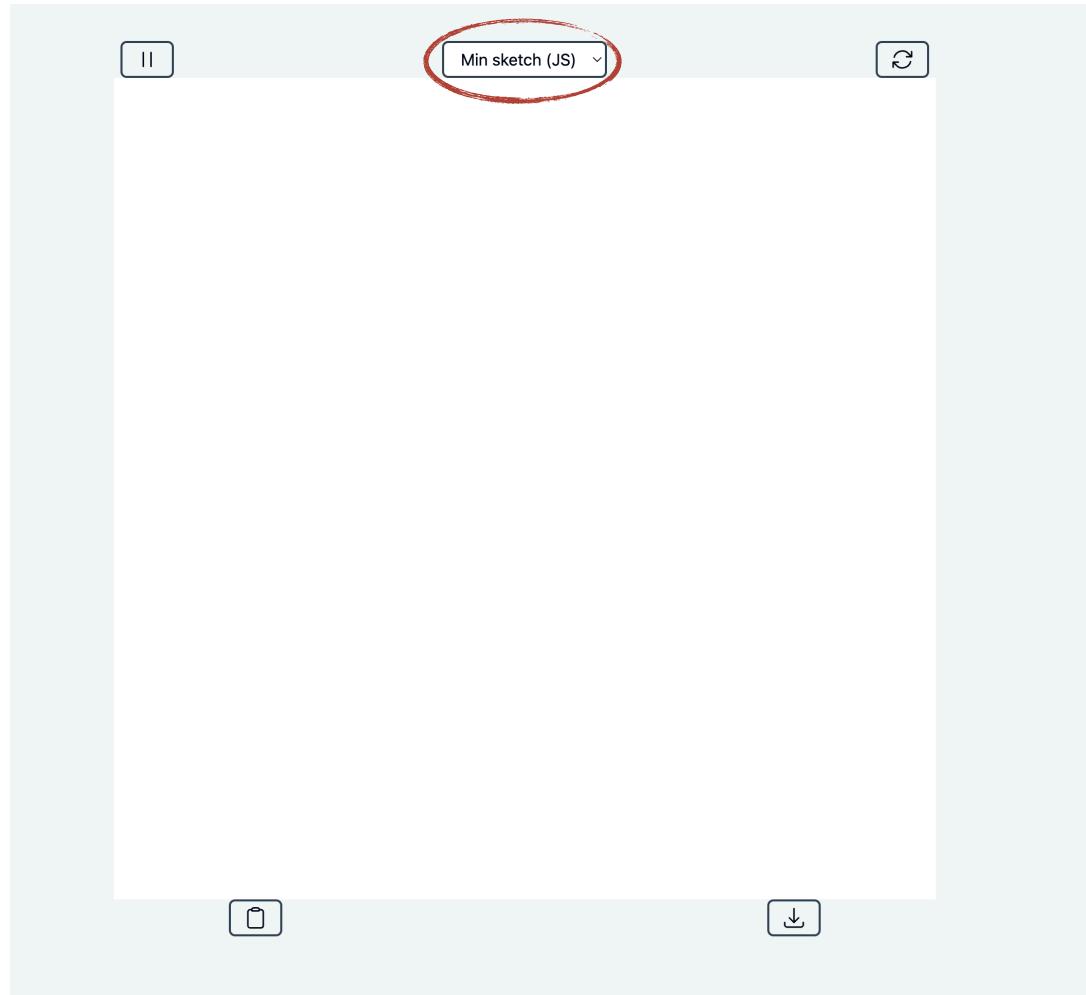
```
1▼ function setup() {  
2    createCanvas(400, 400);  
3}  
4  
5▼ function draw() {  
6    background(220);  
7}
```

Forskjeller i syntax

```
import P5 from "p5";  
  
export function example(p5: P5) {  
    p5.setup = () => {  
        p5.createCanvas(400, 400);  
    };  
  
    p5.draw = () => {  
        p5.background(220);  
    };  
}
```

Kopier det fra eksisterende filer!

# Sondres Superkule Github Repo™



```
import { example } from "./example";

export const sketches: Record<string, SketchClosure> = {
  // alle de andre tingene ...
  "Eksempelet fra forrige slide": example,
};
```

# Setup og draw

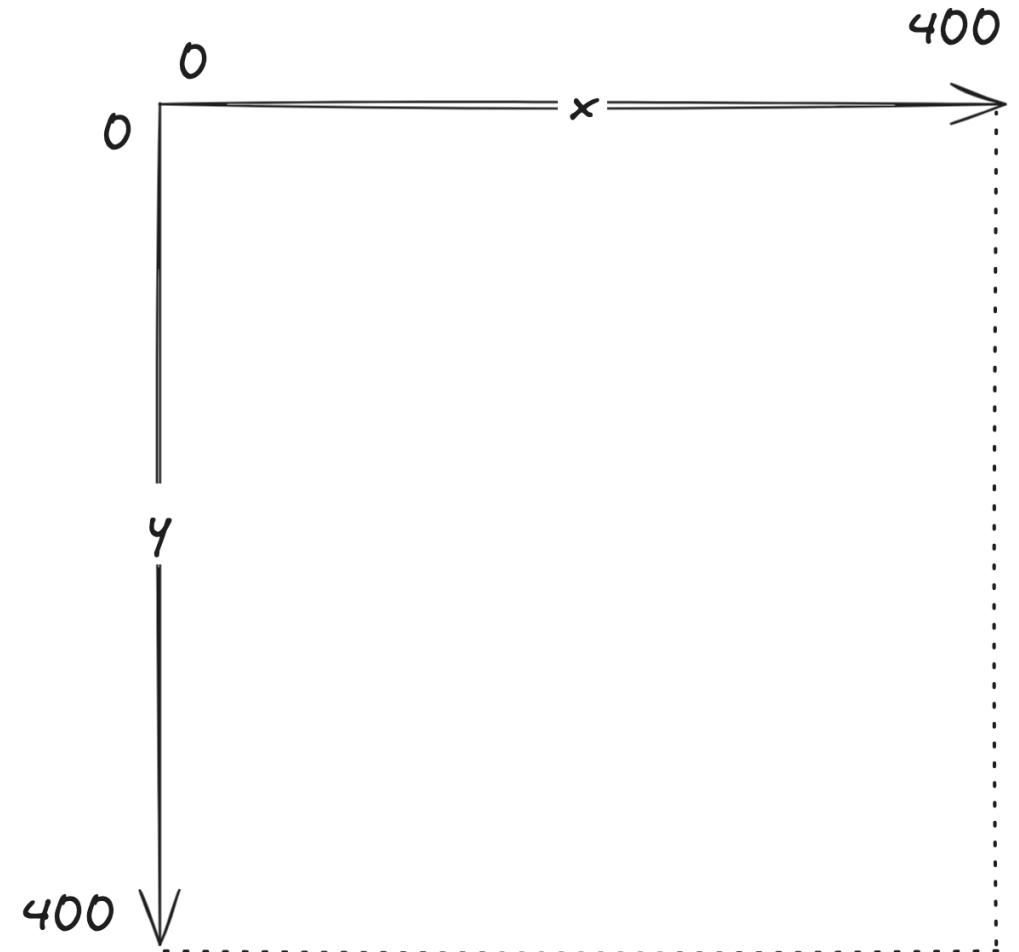
```
import P5 from "p5";  
  
export function example(p5: P5) {  
  p5.setup = () => {  
    p5.createCanvas(400, 400);  
  };  
  
  p5.draw = () => {  
    p5.background(220);  
  };  
}
```

setup()  
↓  
draw()  
X

p5.noLoop()

# Canvas og koordinatsystem

```
p5.createCanvas(400, 400);  
x mot høyre, y nedover
```



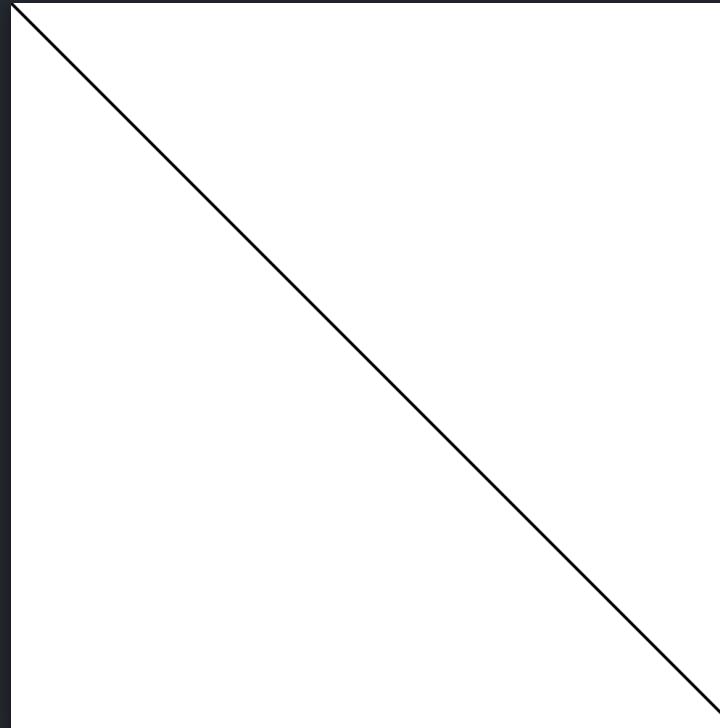
# Linje

```
import P5 from "p5";

export function example(p5: P5) {
  const CANVAS = 500;

  p5.setup = () => {
    p5.createCanvas(CANVAS, CANVAS);
  };

  p5.draw = () => {
    p5.noLoop();
    p5.background("white");
    p5.stroke("black");
    p5.strokeWeight(2);
    p5.line(0, 0, CANVAS, CANVAS);
  };
}
```

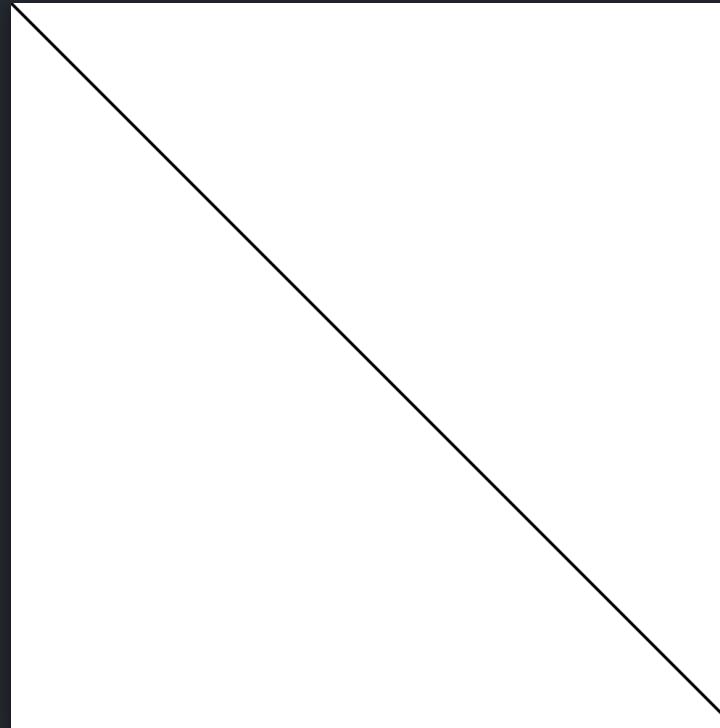


# Linje

```
p5.line(0, 0, CANVAS, CANVAS);

p5.line(xFrom, yFrom, xTo, yTo);

const left = 0;
const right = CANVAS;
const top = 0;
const bottom = CANVAS;
p5.line(left, top, right, bottom);
```



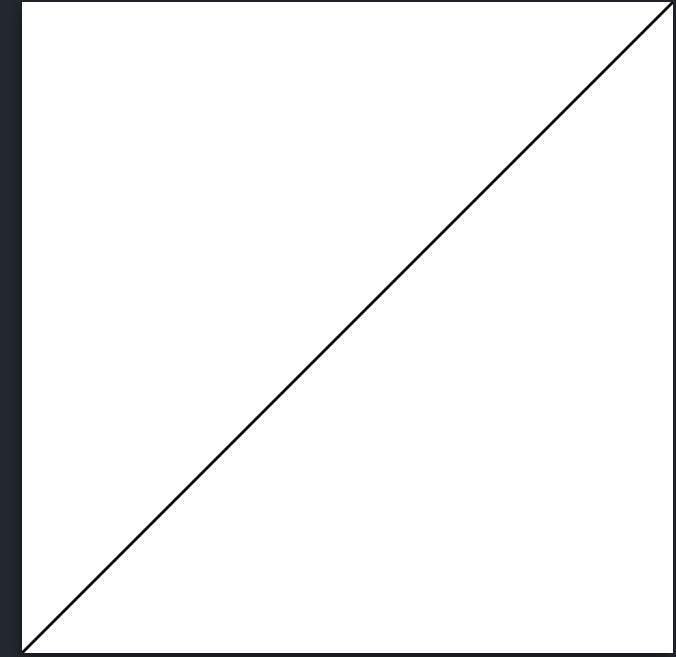
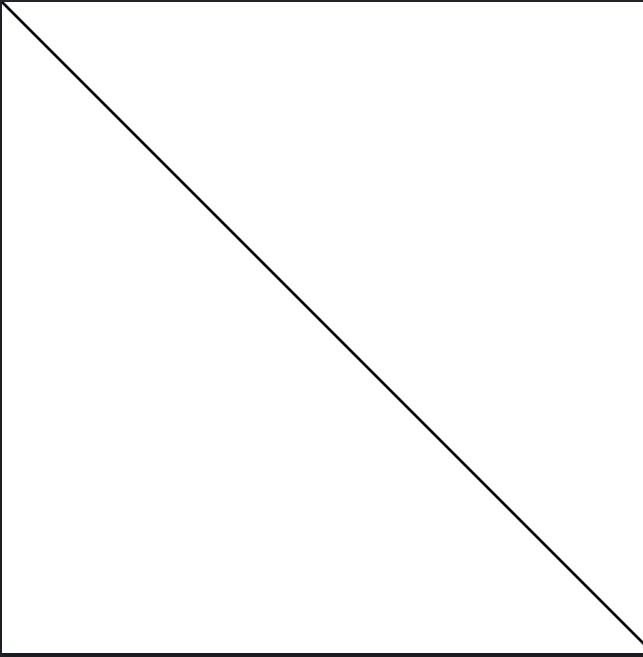
# Linje

Var dette generativ kunst?

Ny idé: 50/50 om vi tegner linja den ene veien eller den andre!

# Linje

```
p5.draw = () => {
  p5.noLoop();
  p5.background("white");
  p5.stroke("black");
  p5.strokeWeight(2);
  const left = 0;
  const right = CANVAS;
  const top = 0;
  const bottom = CANVAS;
  if (p5.random() > 0.5) {
    p5.line(left, top, right, bottom);
  } else {
    p5.line(right, top, left, bottom);
  }
};
```



# Repetisjon / tiling

Hva hvis vi tegnet det flere ganger?

Strategi: flislegging.

Deler opp canvas i flere små fliser

Gjentar det vi gjorde i forrige eksempel i hver flis

# Repetisjon / tiling

```
import P5 from "p5";

export function tilingExample(p5: P5) {
    const CANVAS = 500;
    const tiles = 2;
    const tileSize = CANVAS / tiles;

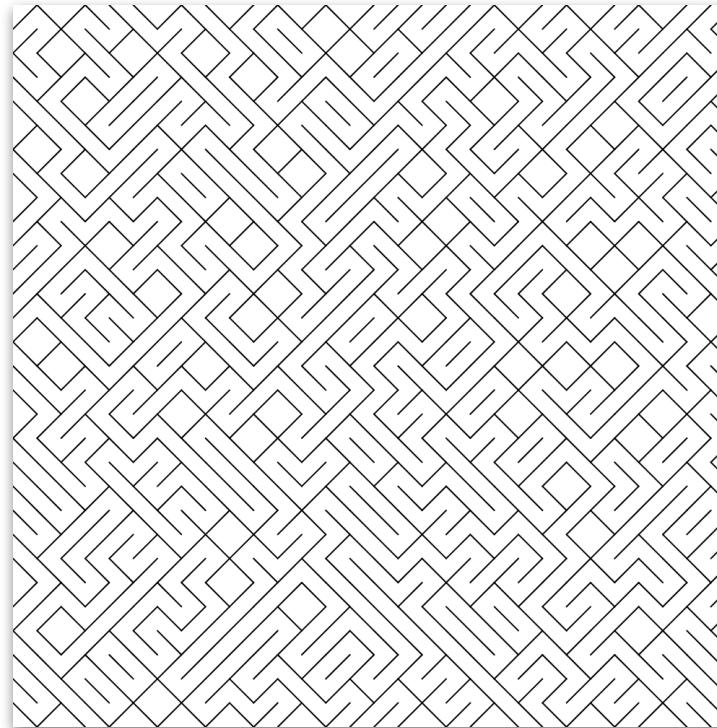
    p5.setup = () => {
        p5.createCanvas(CANVAS, CANVAS);
    };

    p5.draw = () => {
        p5.noLoop();
        p5.background("white");
        p5.stroke("black");
        p5.strokeWeight(2);

        for (let row = 0; row < tiles; row++) {
            for (let column = 0; column < tiles; column++) {
                drawTile(row, column);
            }
        }
    };

    function drawTile(row: number, column: number) {}
}
```

# Repetisjon / tiling



# Støy og tilfeldigheter

Hva var greia med p5.`random()`?

Kan vi bruke den til noe annet?

Idé: tegn en linje, men forskyve den

# Støy og tilfeldigheter

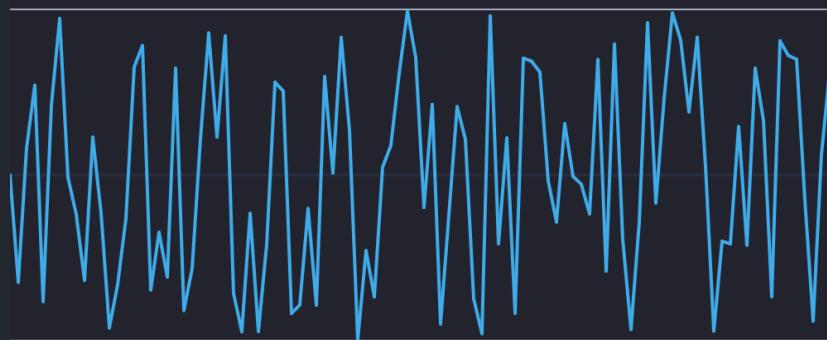
```
p5.draw = () => {
  // forskyver canvas slik at y=0 er midt på
  p5.translate(0, CANVAS / 2);

  const numberOfSections = 100;
  const sectionLength = CANVAS / numberOfSections;
  const amplitude = 100;
  let previousX = 0;
  let previousY = 0;
  for (let i = 0; i < numberOfSections; i++) {
    const newX = previousX + sectionLength;
    const newY = p5.random(-amplitude, amplitude);
    p5.line(previousX, previousY, newX, newY);
    previousX = newX;
    previousY = newY;
  }
};
```



# Støy og tilfeldigheter

```
const newY = p5.random(-amplitude, amplitude);
```



# Støy og tilfeldigheter

```
const newY = p5.randomGaussian(0, amplitude / 2);
```

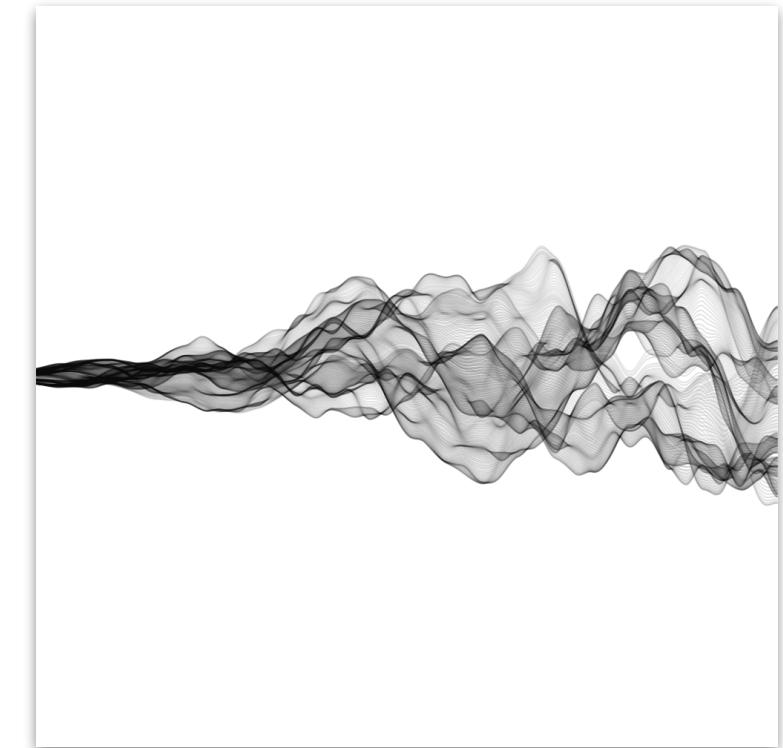
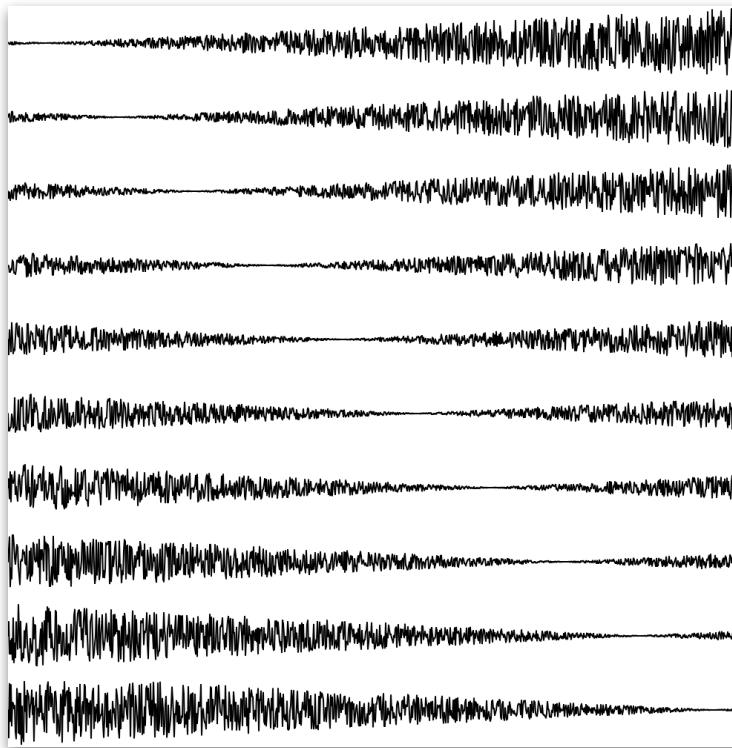
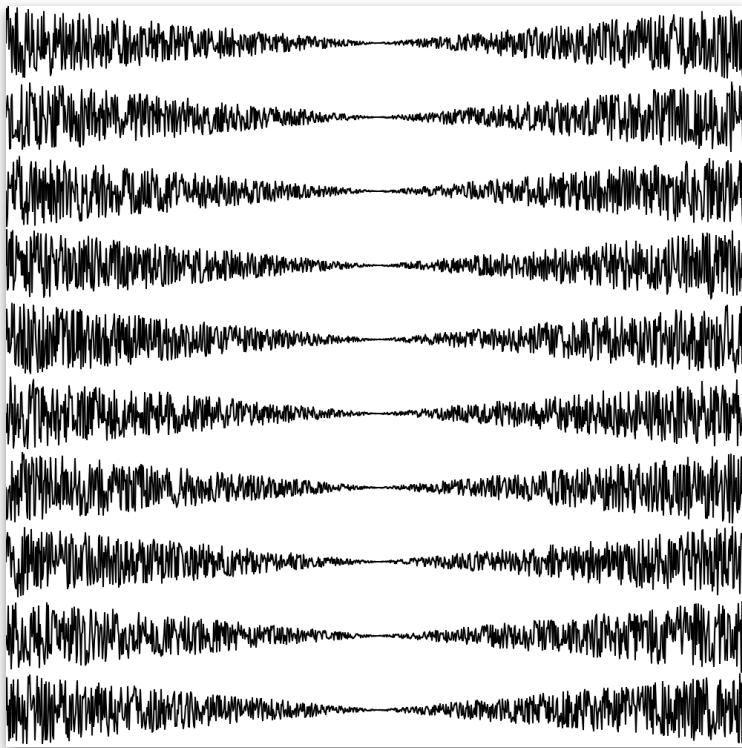


# Støy og tilfeldigheter

```
const noise = p5.noise(newX * 0.01);
const newY = p5.map(noise, 0, 1, -amplitude, amplitude);
```



# Støy og tilfeldigheter

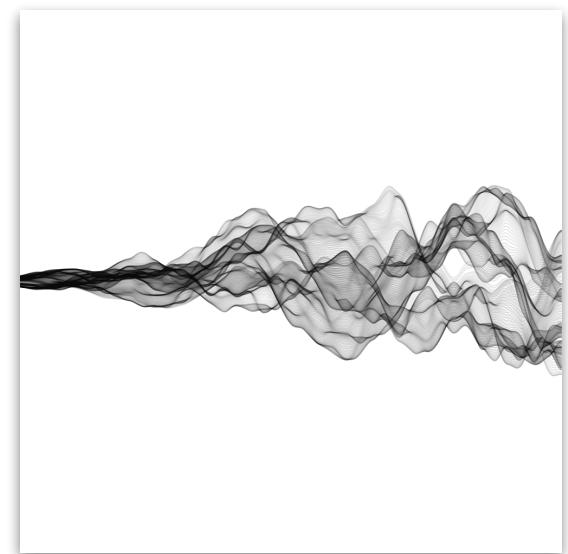
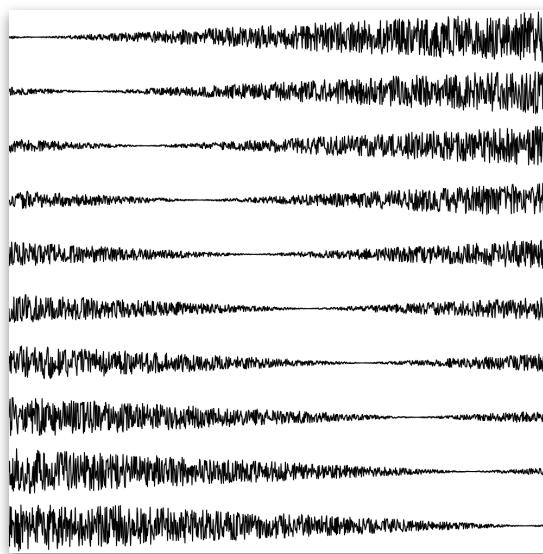
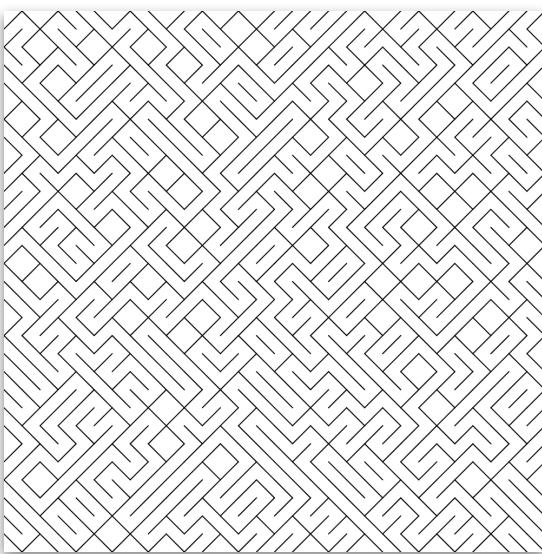


# Verktøykassa og galleriet

Linjer

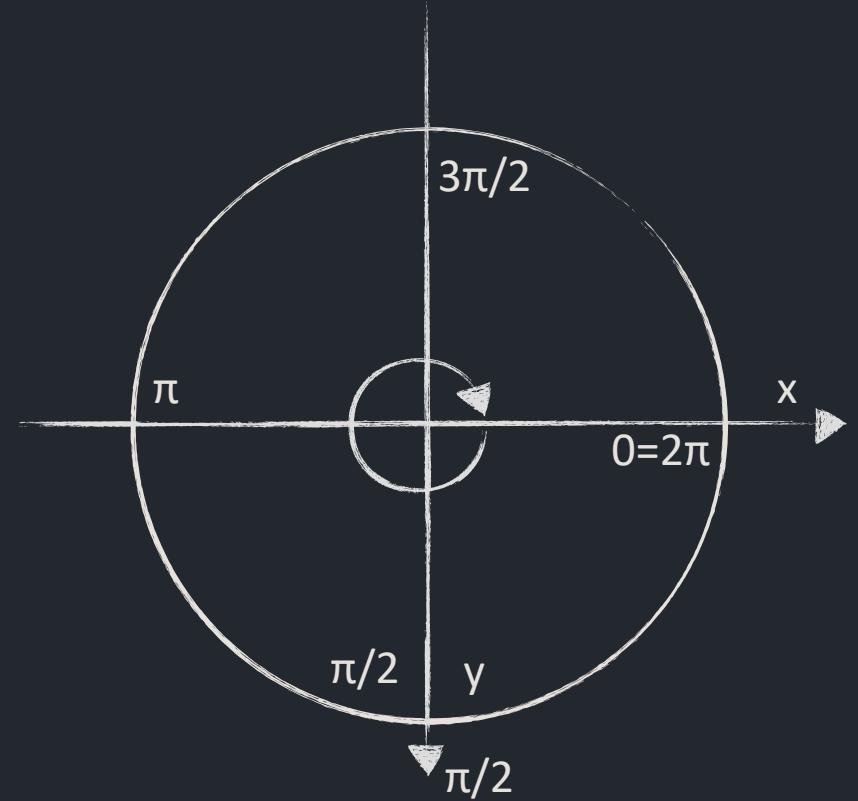
Flislegging

Støy



# Buer

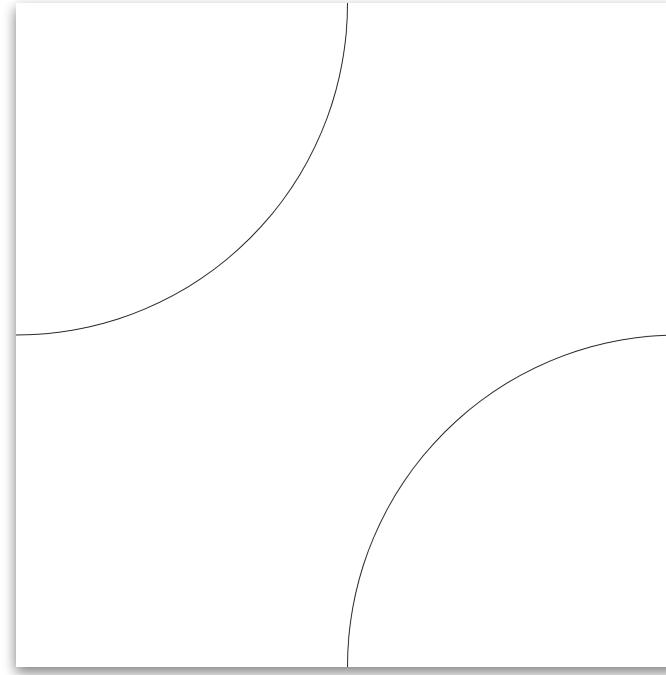
```
p5.arc(  
    ellipseCenterX, // x-koordinat til senteret av ellipsen  
    ellipseCenterY, // y-koordinat til senteret av ellipsen  
    ellipseWidth, // bredden til ellipsen, hvis du hadde tegnet hele  
    ellipseHeight, // høyden til ellipsen, hvis du hadde tegnet hele  
    startAngle, // startvinkelen til buen, i radianer  
    endAngle // sluttvinkelen til buen, i radianer  
);
```



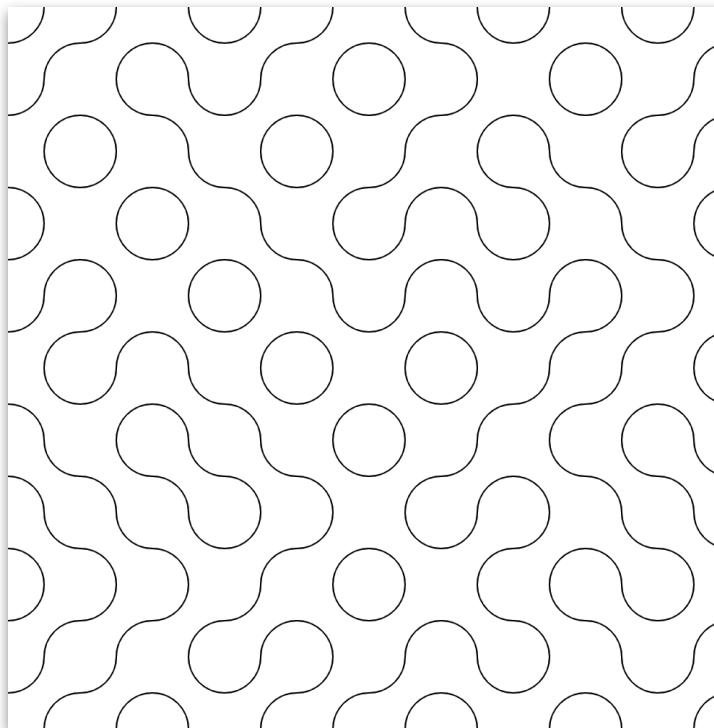
# Buer

```
function drawQuarterCircle(  
    circleCenterX: number,  
    circleCenterY: number,  
    circleRadius: number,  
    startAngle: number  
) {  
    p5.arc(  
        circleCenterX,  
        circleCenterY,  
        circleRadius * 2,  
        circleRadius * 2,  
        startAngle,  
        startAngle + p5.HALF_PI  
    );  
}  
  
const left = 0;  
const right = CANVAS;  
const top = 0;  
const bottom = CANVAS;  
if (p5.random() > 0.5) {  
    drawQuarterCircle(left, bottom, CANVAS / 2, p5.PI + p5.HALF_PI);  
    drawQuarterCircle(right, top, CANVAS / 2, p5.HALF_PI);  
} else {  
    drawQuarterCircle(left, top, CANVAS / 2, 0);  
    drawQuarterCircle(right, bottom, CANVAS / 2, p5.PI);  
}
```

Buer



Buer

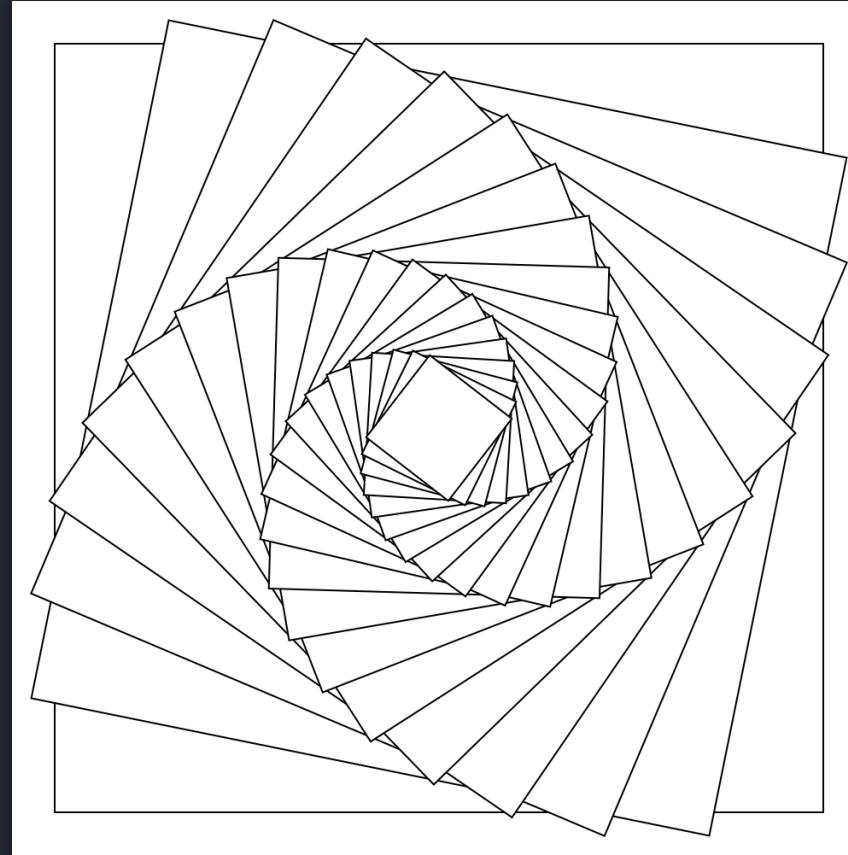


# Former

```
p5.quad // firkantede polygoner  
p5.rect // rektangel  
p5.square // kvadrat  
  
p5.ellipse // ellipse  
p5.circle // sirkel  
p5.arc // del av ellipse  
  
p5.triangle // trekant  
  
p5.beginShape // starter en figur  
p5.vertex // legger til et punkt i en figur  
p5.endShape // avslutter en figur
```

# Former

```
p5.draw = () => {
  p5.noLoop();
  let size = CANVAS * 0.9;
  p5.translate(CANVAS / 2, CANVAS / 2);
  p5.rectMode(p5.CENTER);
  for (let i = 0; i < repeats; i++) {
    p5.square(0, 0, size);
    p5.rotate(angle);
    size = size * sizeScaling;
  }
};
```



# Verktøykassa og galleriet

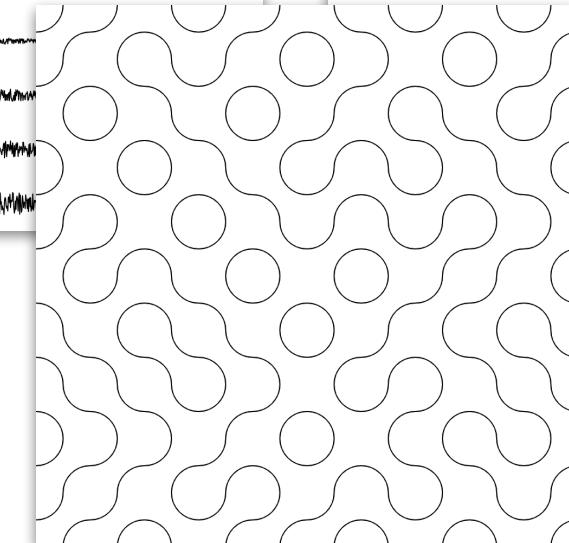
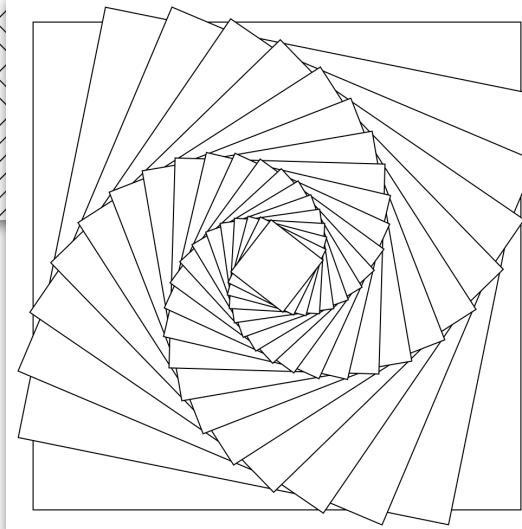
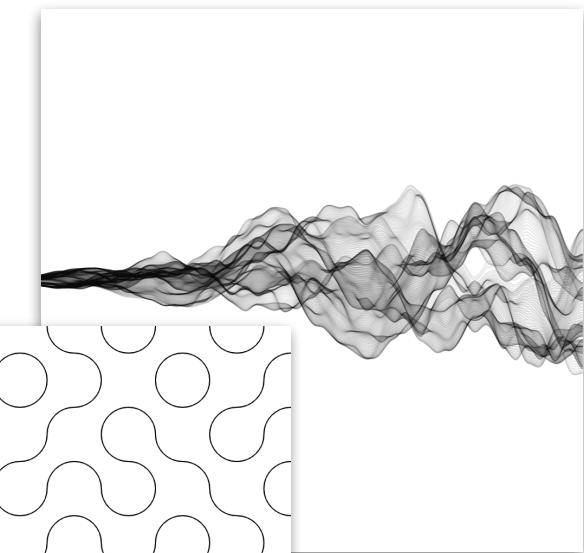
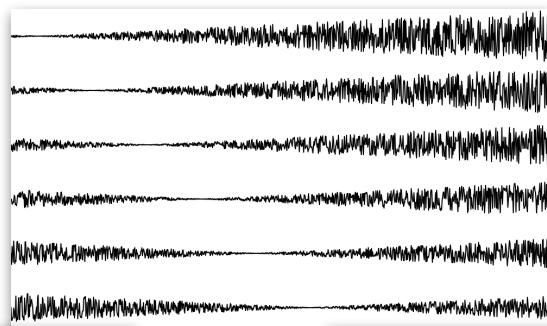
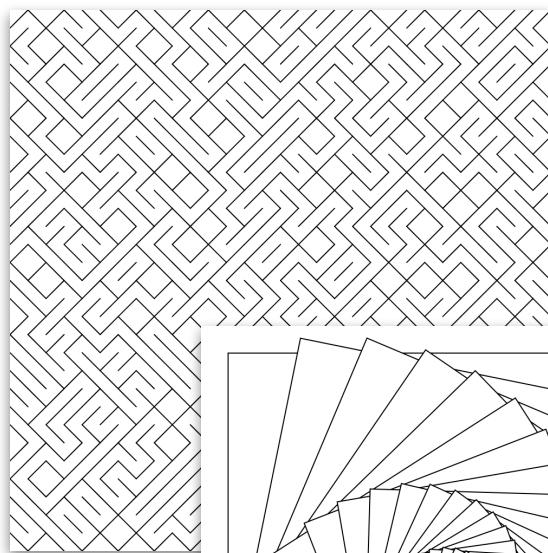
Linjer

Flislegging

Støy

Buer

Former



# Farger

```
p5.background("black");
p5.fill("black");
p5.stroke("black");

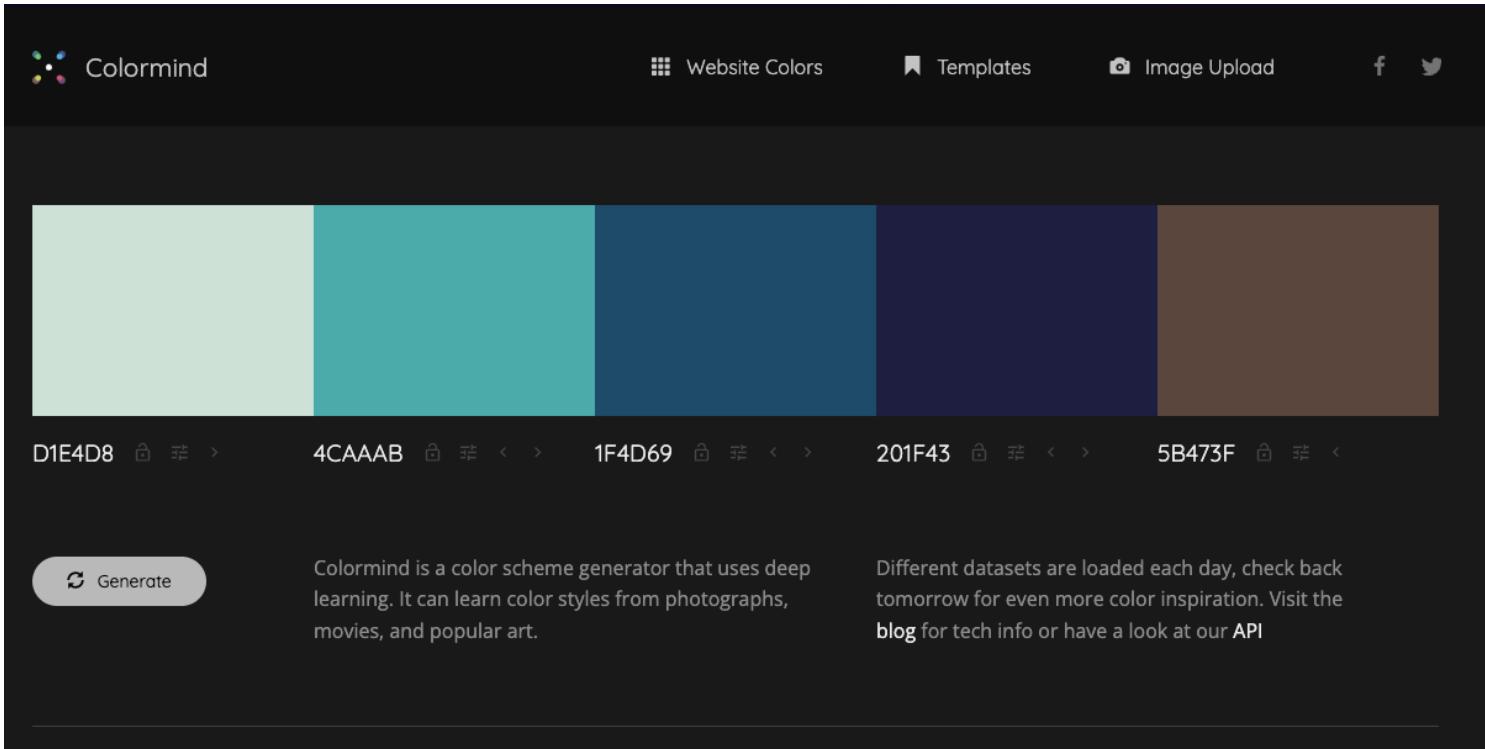
const color = p5.color("black");
p5.background(color);
p5.fill(color);
p5.stroke(color);

p5.color("black"); // CSS-farger
p5.color("#000000"); // Hex-kode
p5.color(0); // Gråtone, 0-255
p5.color(0, 0, 0); // RGB, 0-255
p5.color(0, 0, 0, 255); // RGBA, 0-255

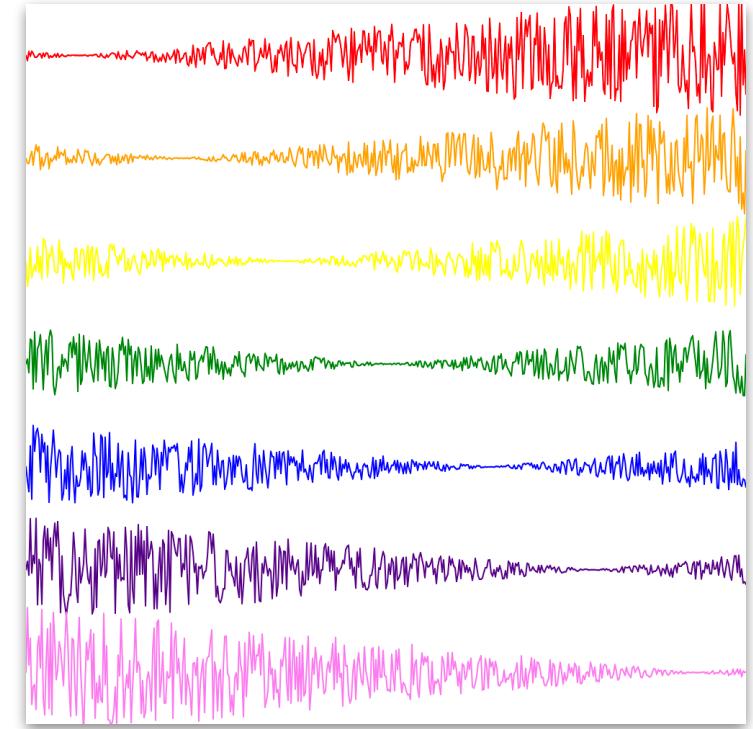
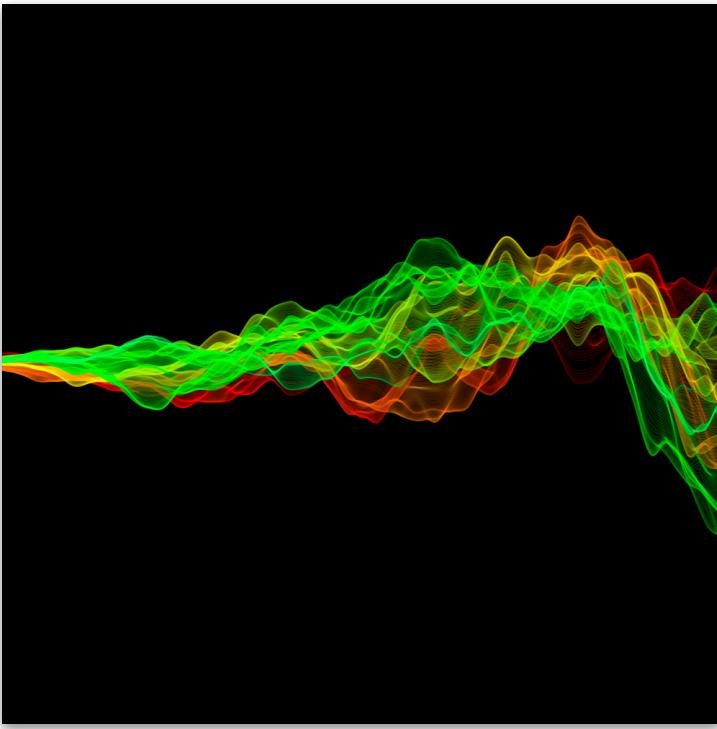
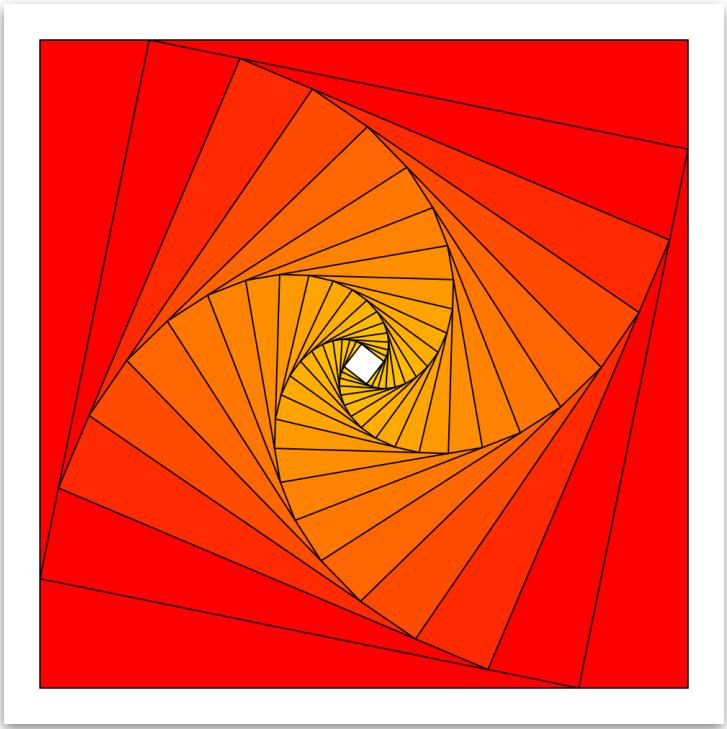
p5.colorMode(p5.HSL);
p5.lerpColor(color1, color2, 0.5); // 50% mellom color1 og color2
```

# Farger

Vansklig å generere bra paletter selv, gå for “håndplukket”



# Farger



# Tid og bevegelse

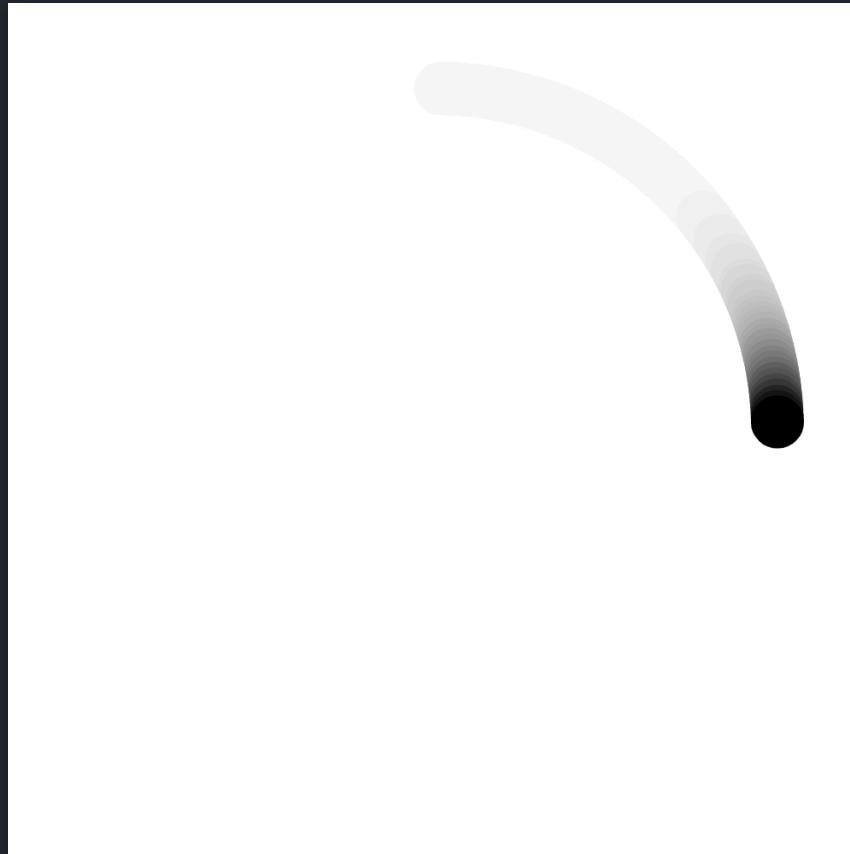
```
export function timeExample(p5: P5) {
  const CANVAS = 500;
  let timeInSeconds = 0;

  p5.setup = () => {
    p5.createCanvas(CANVAS, CANVAS);
    p5.background("white");
  };

  p5.draw = () => {
    timeInSeconds += p5.deltaTime / 1_000;
    p5.background(255, 20);

    p5.translate(CANVAS / 2, CANVAS / 2);
    p5.rotate(timeInSeconds);

    p5.fill("black");
    p5.circle(0, 200, 30);
  };
}
```



# Verktøykassa og galleriet

Linjer

Flislegging

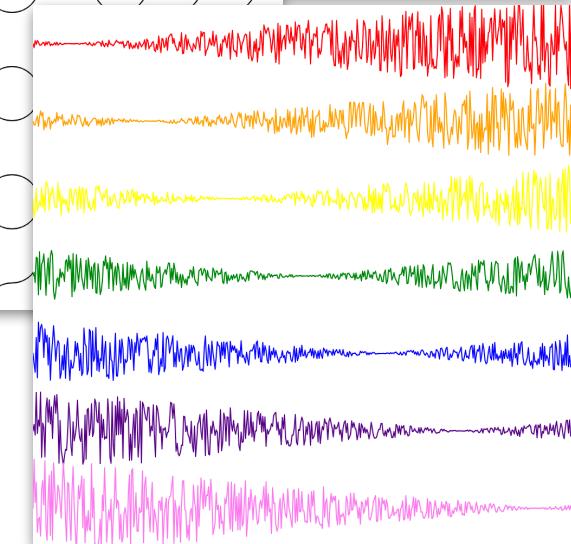
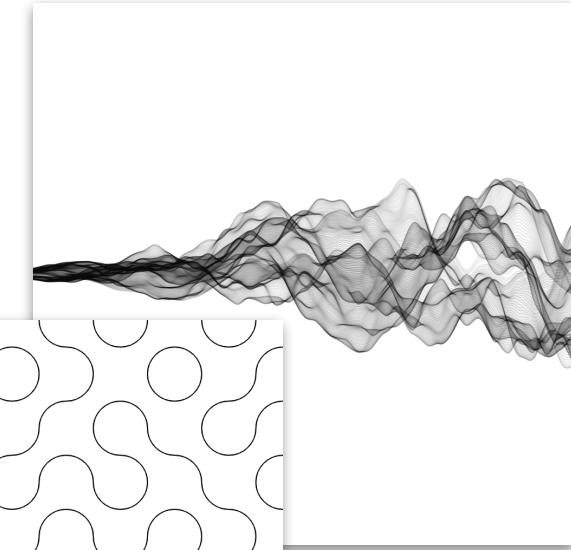
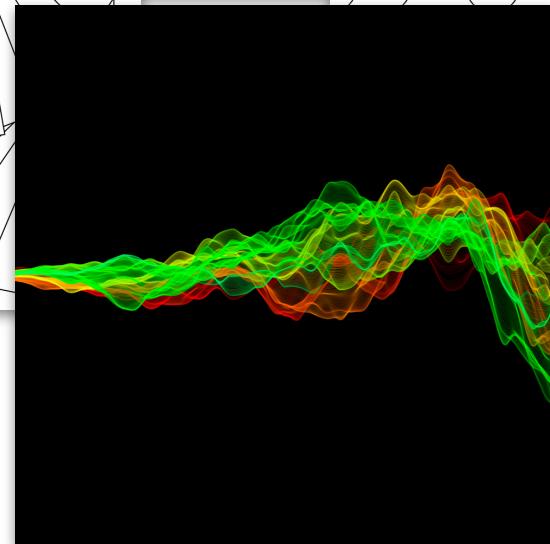
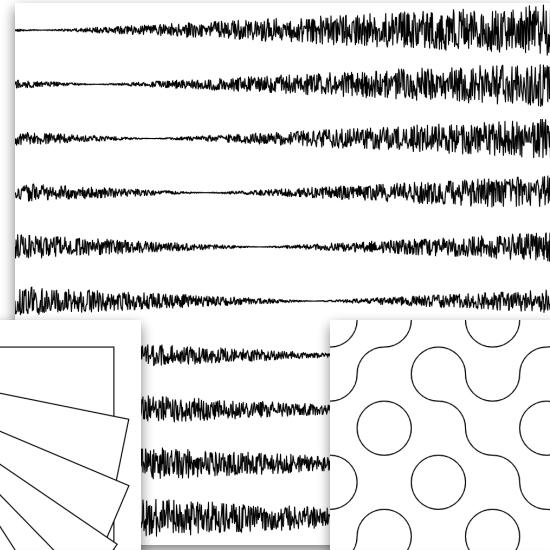
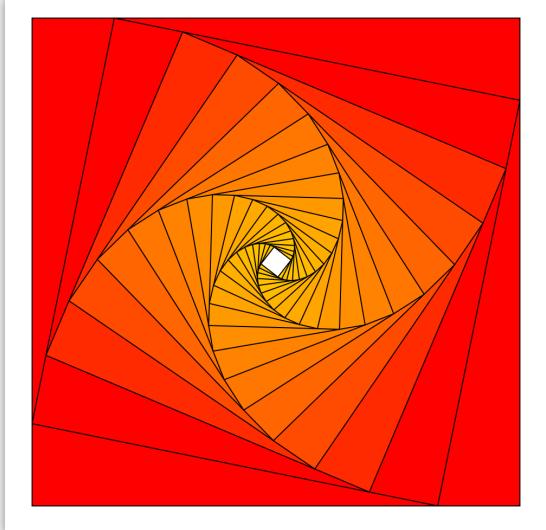
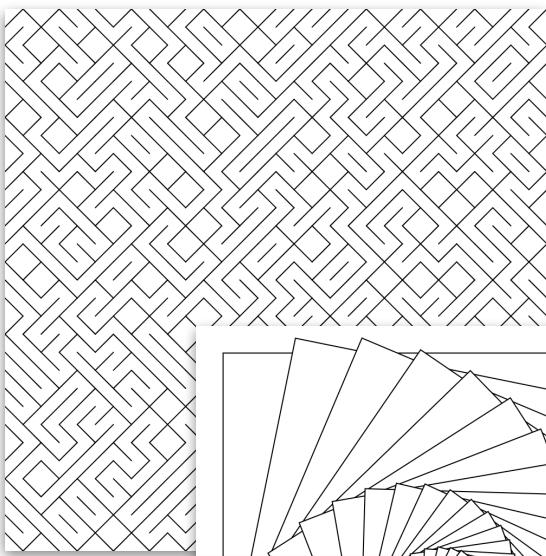
Støy

Buer

Former

Farger

Tid



# Mer

Interagere med musepeker med p5.mouseX og p5.mouseY

Typografi

Bezier-kurver

3D

Hva skjer nå?

# Lenker

p5.js web editor:

<https://editor.p5js.org/>

Sondres Superkule Github Repo™ + slides

<https://github.com/bekk/p5-gen-art-workshop>

p5.js dokumentasjon

<https://p5js.org/reference/>

p5.js eksempler

<https://p5js.org/examples/>