

10. MINTwoch

Programmiere dein eigenes Kunstwerk

Wie aus Code Bilder werden

07.07.2021

```
// All the paths
let paths = [];
// Are we painting?
let painting = false;
// How long until the next circle
let next = 0;
// Where are we now and where were we?
let current;
let previous;

function setup() {
  createCanvas(720, 400);
  current = createVector(0,0);
  previous = createVector(0,0);
};

function draw() {
  background(200);

  // If it's time for a new point
  if (millis() > next && painting) {

    // Grab mouse position
    current.x = mouseX;
    current.y = mouseY;

    // New particle's force is based on mouse movement
    let force = p5.Vector.sub(current, previous);
    force.mult(0.05);

    // Add new particle
    paths[paths.length - 1].add(current, force);

    // Schedule next circle
    next = millis() + random(100);

    // Store mouse values
    previous.x = current.x;
    previous.y = current.y;
  }

  // Draw all paths
  for( let i = 0; i < paths.length; i++) {
    paths[i].update();
    paths[i].display();
  }
}
```

Zeitplan

- 01** Begrüßung
- 02** Einführung
- 03** Beispiel 1: Einfache Formen
- 04** Beispiel 2: Kaleidoskop
- 05** Beispiel 3: Wavemaker
- 06** Eigener Sketch
- 07** Fragen

Team



Cansu Demir



Vivien Wallner



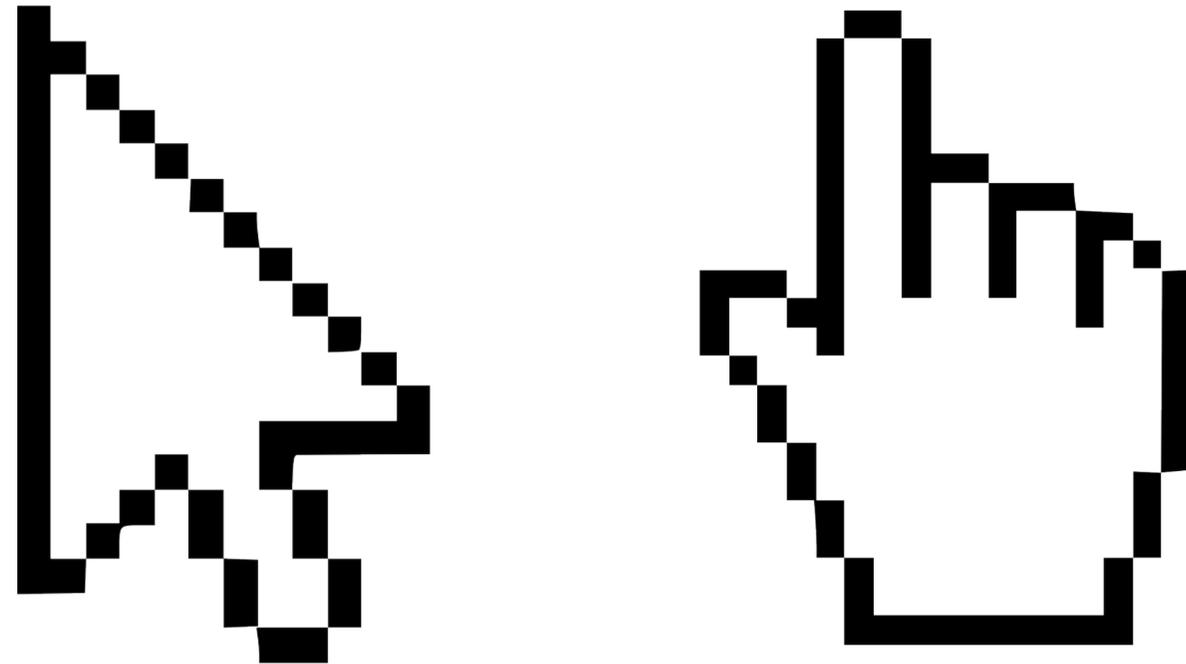
Anne Münzner

Pixel



Pixel

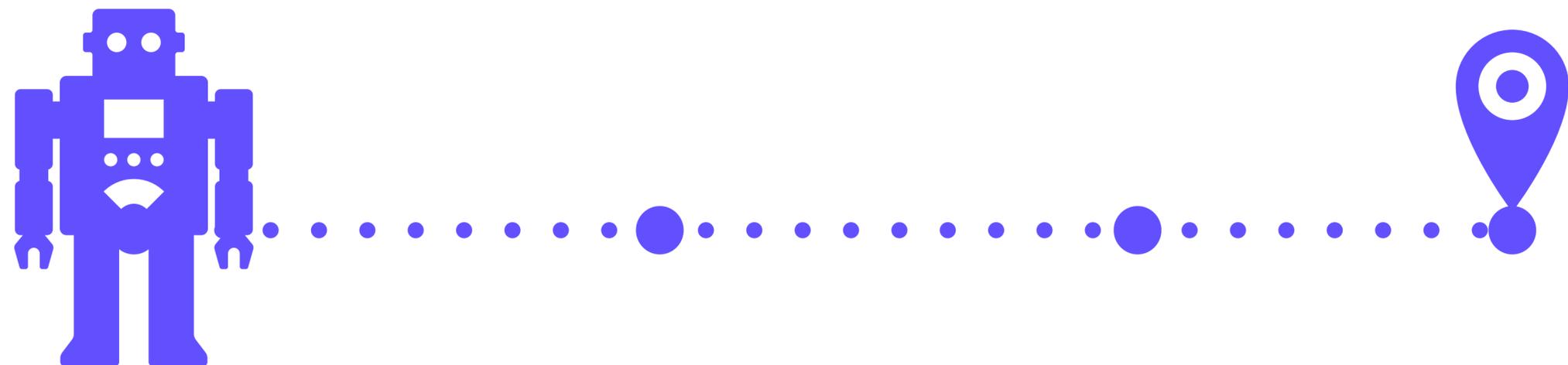
- Abkürzung Pixel Elemente (Bild-Teil oder Bildpunkt)
- Kleinste Einheit die auf Bildschirm dargestellt wird
- Digitale Bilder sind durch viele einzelne Pixel zusammengesetzt



Algorithmen

Genauere Anleitung und Anweisung, um ein Ziel zu erreichen / Problem zu lösen

- Kann auch analog sein (z.B. ein Rezept, eine Bauanleitung)
- Algorithmen werden als Code nach genauem Syntax geschrieben, um Programme / Software zu erstellen



Computercode

... kann durch eine Maschine ausgeführt werden

... ist in einer bestimmten Programmiersprache geschrieben (z.B Java, Python, C++)

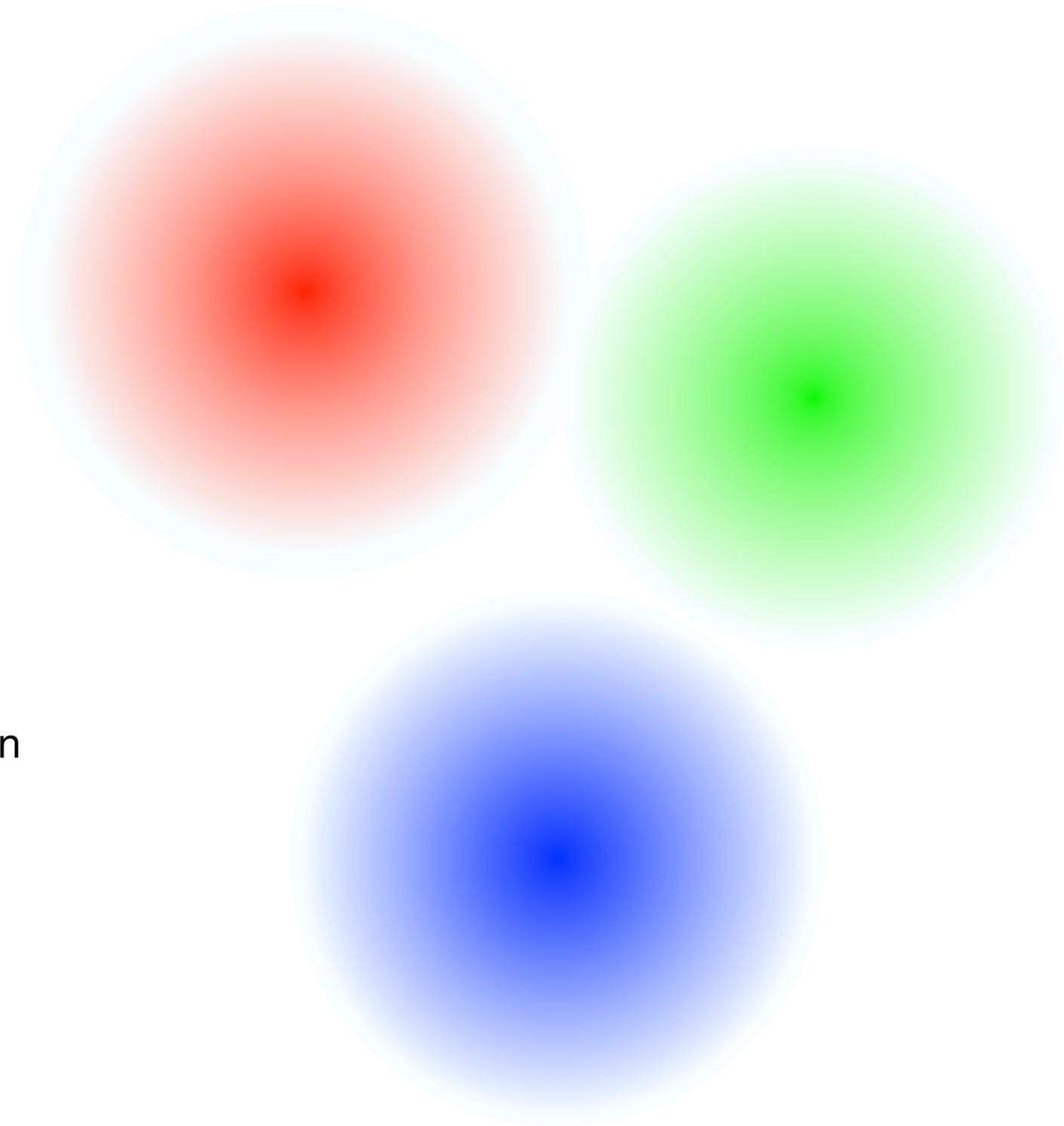
Dabei muss der Code genau im Syntax der gewählten Programmiersprache geschrieben werden, damit die Maschine ihn versteht.

; Trennung von einzelnen Befehlen

// Kommentare, die vom Computer nicht
verarbeitet werden sollen

RGB

- **Rot, Grün, Blau**
- beliebige Farben können aus den 3 Farben gemischt werden
- `rgb(130, 105, 159)`
- `fill(r,g,b) —> fill(130,105,159) -> füllt ein Objekt mit der oben gezeigten Farbe`



RGB

rgb(130, 105, 159)

RGB Calculator

rgb(115, 81, 255)

#7351ff

hsl(252, 100%, 66%)

R: 115

G: 81

B: 255

[Use this color in our Color Picker](#)

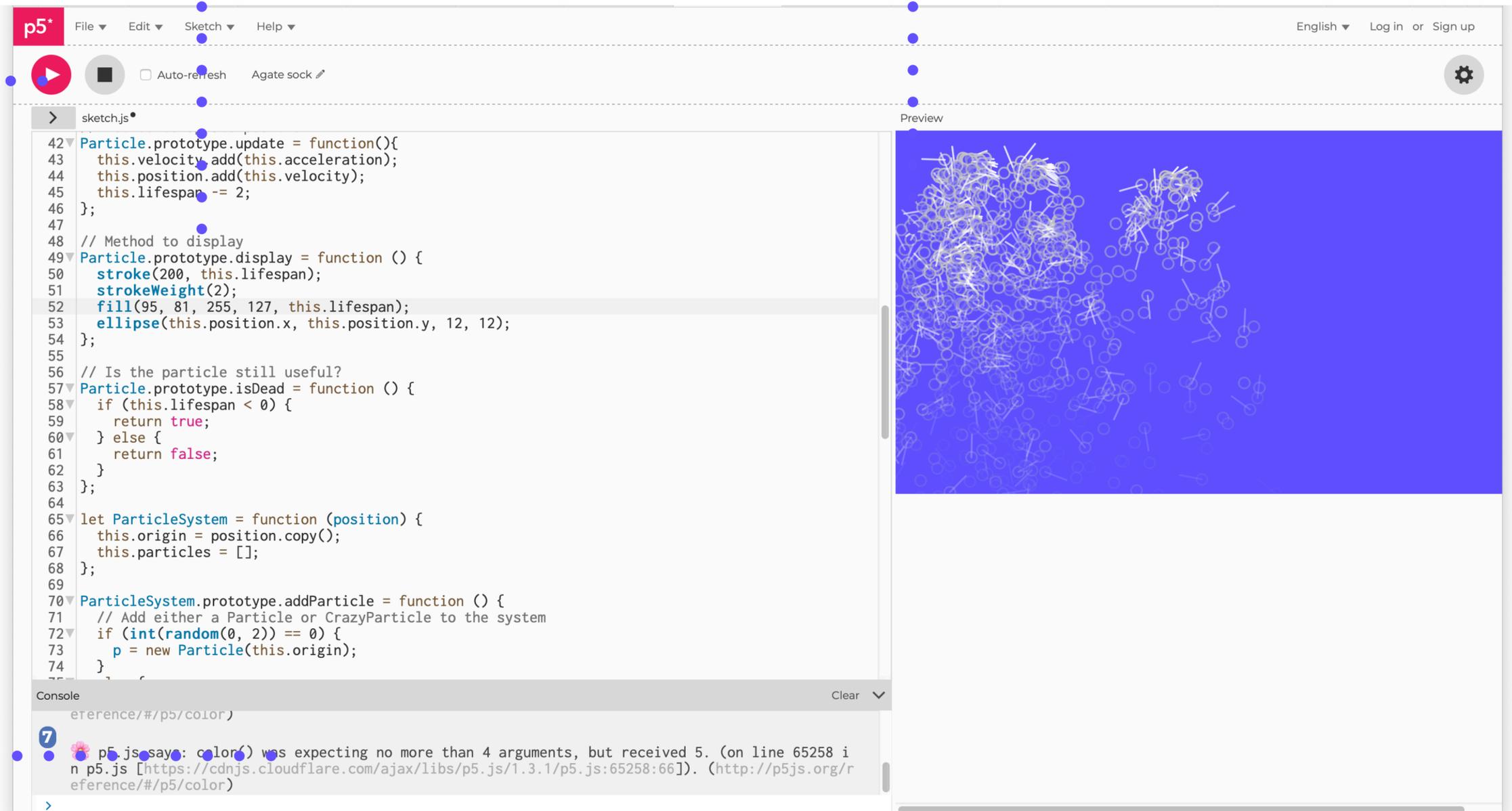
p5.js Online editor

- **Code Editor**
- (Code hier Eingeben)

- **Vorschau**
- (Ergebnis hier sehen & testen)

- **Play Button**
- (Code ausführen)

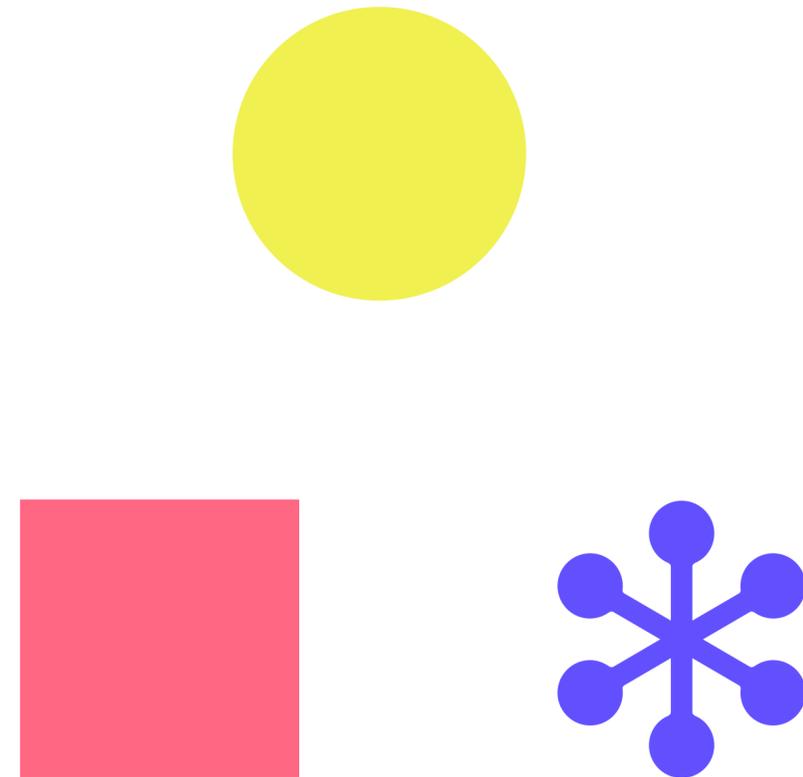
- **Konsole**
- (Fehler hier einsehen)



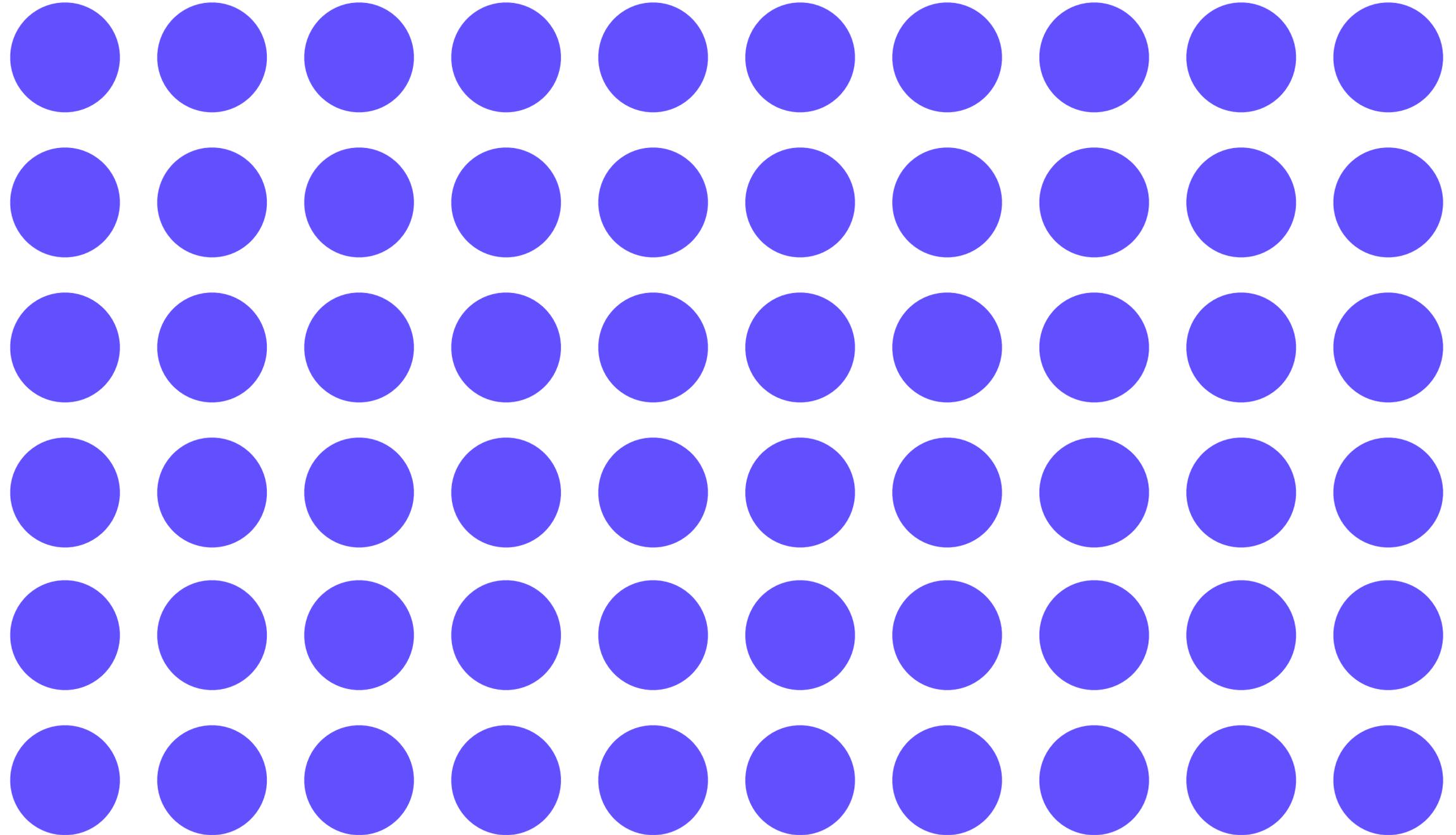
Beispiel 01

Einfache Formen

```
function setup() {  
  // Create the canvas  
  createCanvas(600, 600);  
  background(255);  
  
  // A rectangle  
  fill(255, 51, 51);  
  stroke(255, 51, 51);  
  rect(80, 300, 120, 120);  
  
  // An ellipse  
  fill(255, 255, 102);  
  noStroke();  
  ellipse(300, 120, 150, 150);  
  strokeWeight(15);  
  stroke(102, 102, 255);  
  
  // A design for a simple flower  
  translate(450, 350);  
  noStroke();  
  fill(102, 102, 255);  
  for (let i = 0; i < 6; i++) {  
    ellipse(0, 60, 30, 30);  
    rotate(PI/3);  
  }  
  
  for (let i = 0; i < 6; i++){  
    rectMode(RADIUS);  
    rect(0, 0, 6, 60);  
    rotate(PI/3);  
  }  
}
```



**Was können
Computer
besser als wir?**



Beispiel 02

Kaleidoskop

```
// Symmetry corresponding to the number of reflections. Change the num
let symmetry = 6;

let angle = 360 / symmetry;
let saveButton, clearButton, mouseButton, keyboardButton;
let slider;

function setup() {
  createCanvas(710, 710);
  angleMode(DEGREES);
  background(127);

  // Creating the save button for the file
  saveButton = createButton('save');
  saveButton.mousePressed(saveFile);

  // Creating the clear screen button
  clearButton = createButton('clear');
  clearButton.mousePressed(clearScreen);

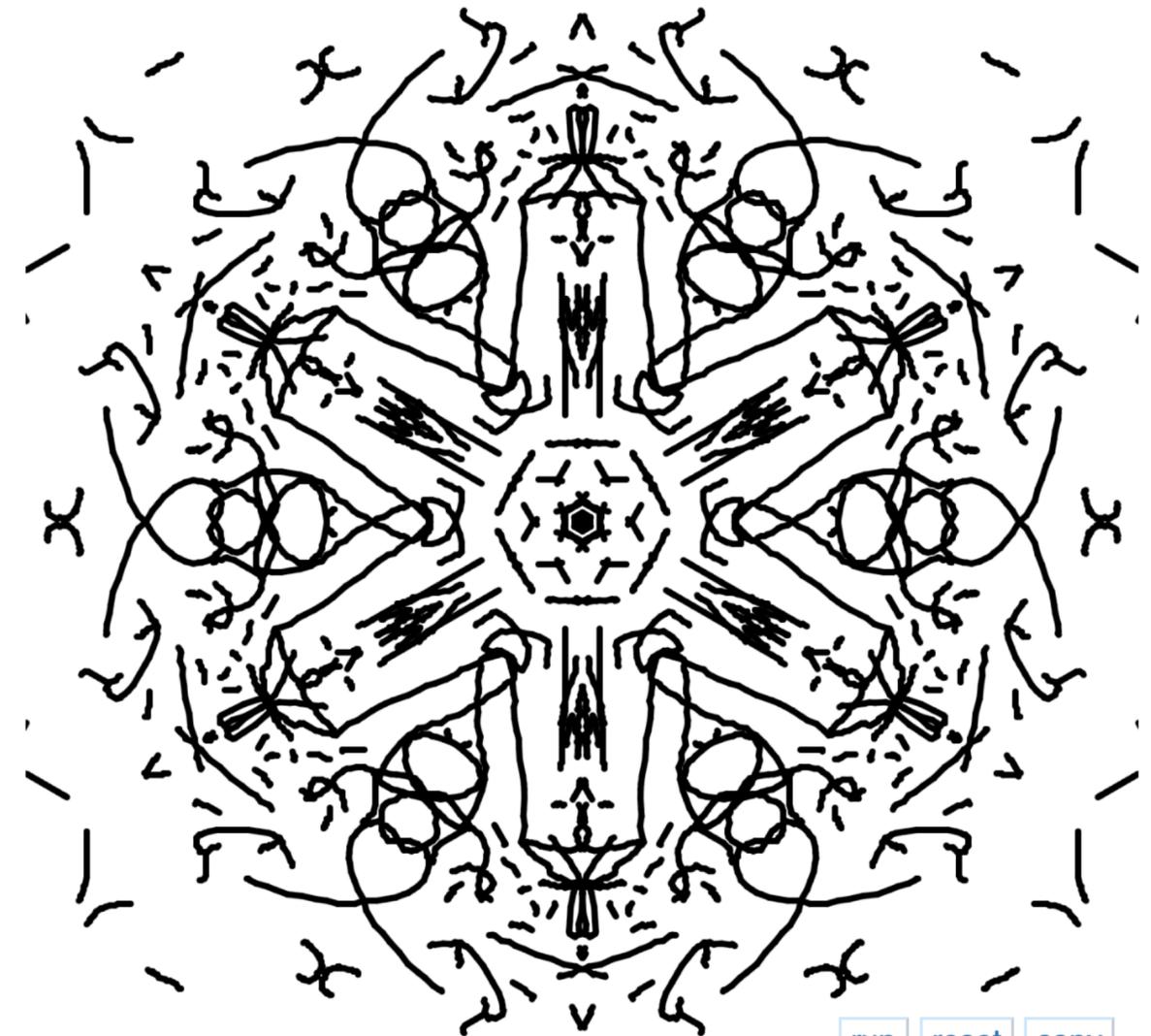
  // Creating the button for Full Screen
  fullscreenButton = createButton('Full Screen');
  fullscreenButton.mousePressed(screenFull);

  // Setting up the slider for the thickness of the brush
  brushSizeSlider = createButton('Brush Size Slider');
  sizeSlider = createSlider(1, 32, 4, 0.1);
}

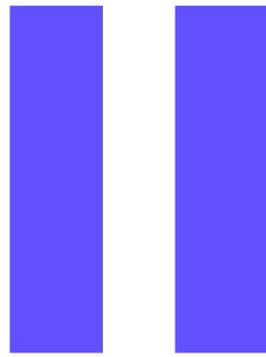
function draw() {
  translate(width / 2, height / 2);

  if (mouseX > 0 && mouseX < width && mouseY > 0 && mouseY < height) {
    let mx = mouseX - width / 2;
    let my = mouseY - height / 2;
    let pmx = pmouseX - width / 2;
    let pmy = pmouseY - height / 2;

    if (mouseIsPressed) {
      for (let i = 0; i < symmetry; i++) {
        rotate(angle);
        let sw = sizeSlider.value();
        strokeWeight(sw);
        line(mx, my, pmx, pmy);
        push();
        scale(1, -1);
        line(mx, my, pmx, pmy);
        pop();
      }
    }
  }
}
```



run reset copy



Beispiel 03

Wavemaker

```
let t = 0; // time variable

function setup() {
  createCanvas(600, 600);
  noStroke();
  fill(95, 81, 255);
}

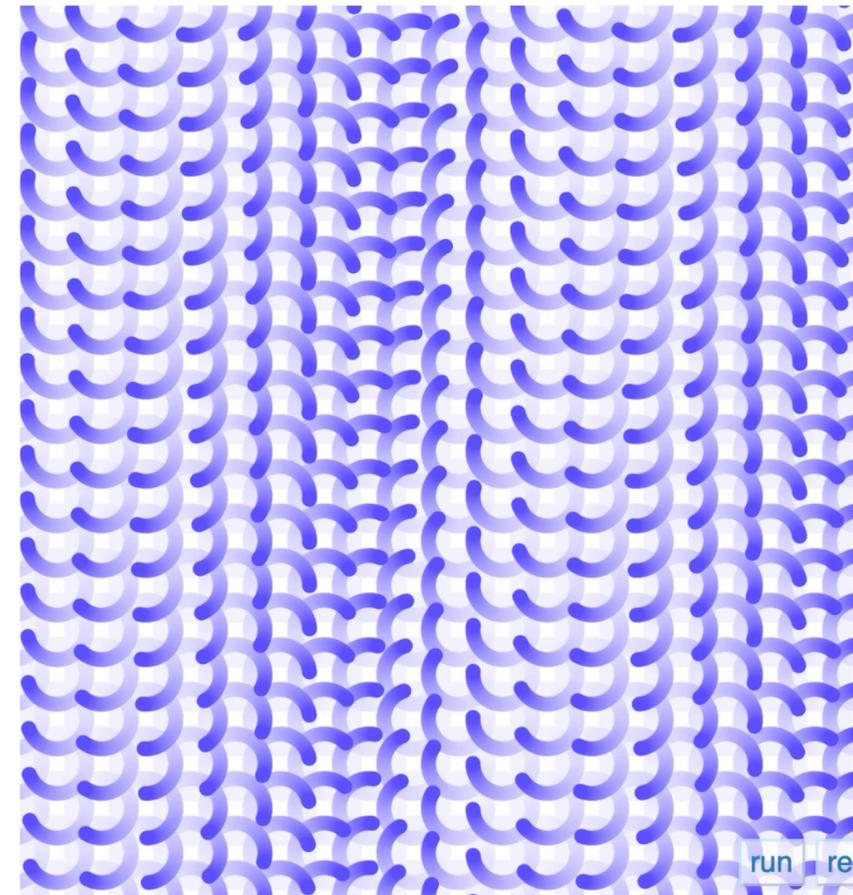
function draw() {
  background(255, 10); // translucent background (creates trails)

  // make a x and y grid of ellipses
  for (let x = 0; x <= width; x = x + 30) {
    for (let y = 0; y <= height; y = y + 30) {
      // starting point of each circle depends on mouse position
      const xAngle = map(mouseX, 0, width, -4 * PI, 4 * PI, true);
      const yAngle = map(mouseY, 0, height, -4 * PI, 4 * PI, true);
      // and also varies based on the particle's location
      const angle = xAngle * (x / width) + yAngle * (y / height);

      // each particle moves in a circle
      const myX = x + 20 * cos(2 * PI * t + angle);
      const myY = y + 20 * sin(2 * PI * t + angle);

      ellipse(myX, myY, 10); // draw particle
    }
  }

  t = t + 0.01; // update time
}
```



run reset copy

Beispiel 04

Dein eigener Name

```
p5* File Edit Sketch Help
▶ Auto-refresh Gold fox
> sketch.js
1 function setup() {
2   createCanvas(800, 800);
3 }
4 function draw() {
5   //background(220);
6   textSize(200);
7   stroke(255,116,135);
8   strokeWeight(10);
9   fill(95, 81, 255, 100);
10  text("Vivien", 150, 320);
11 }
```



Vivien

TODO

- 01** Finde heraus was ein Mausklick bewirkt.
- 02** Ändere das Objekt/den Namen. (Hier ein Bsp. zeigen als Inspiration)
- 03** Ändere das Output bei Mausklick.
- 04** Vergrößere die Leinwand.
- 05** Ändere die Farbe des Objekts.

Kahoot

<https://kahoot.it/>

Codes

- **Kaleidoscope** <https://p5js.org/examples/interaction-kaleidoscope.html>
- **Triangle Strip** <https://p5js.org/examples/form-triangle-strip.html>
- **Drawing** <https://p5js.org/examples/hello-p5-drawing.html>
- **Hue (Color)** <https://p5js.org/examples/color-hue.html>
- **Input/Button** <https://p5js.org/examples/dom-input-and-button.html>
- **Primitive shapes** <https://p5js.org/examples/form-shape-primitives.html>
- **Star** <https://p5js.org/examples/form-star.html>
- **Regular Polygon** <https://p5js.org/examples/form-regular-polygon.html>
- **Pattern** <https://p5js.org/examples/drawing-patterns.html>
- **Flower Pulses** <https://p5js.org/examples/drawing-pulses.html>

Hier gibt es mehr!

- 01** Coding Club Salzburg <https://its.fh-salzburg.ac.at>
- 02** MINT Salzburg <https://www.mint-salzburg.at>
- 03** Computercamp <https://www.computercamp.at>
- 04** The Coding Train <https://thecodingtrain.com>

Center for Human-Computer Interaction

Forschung an der Schnittstelle
von Mensch und Computer

Teil der Universität Salzburg

Design, Programmierung,
Psychologie, Materialien,
Roboter, Games, Mobilität,
Medienkunst

