

expand your toolbox



Introduction
to creative
coding



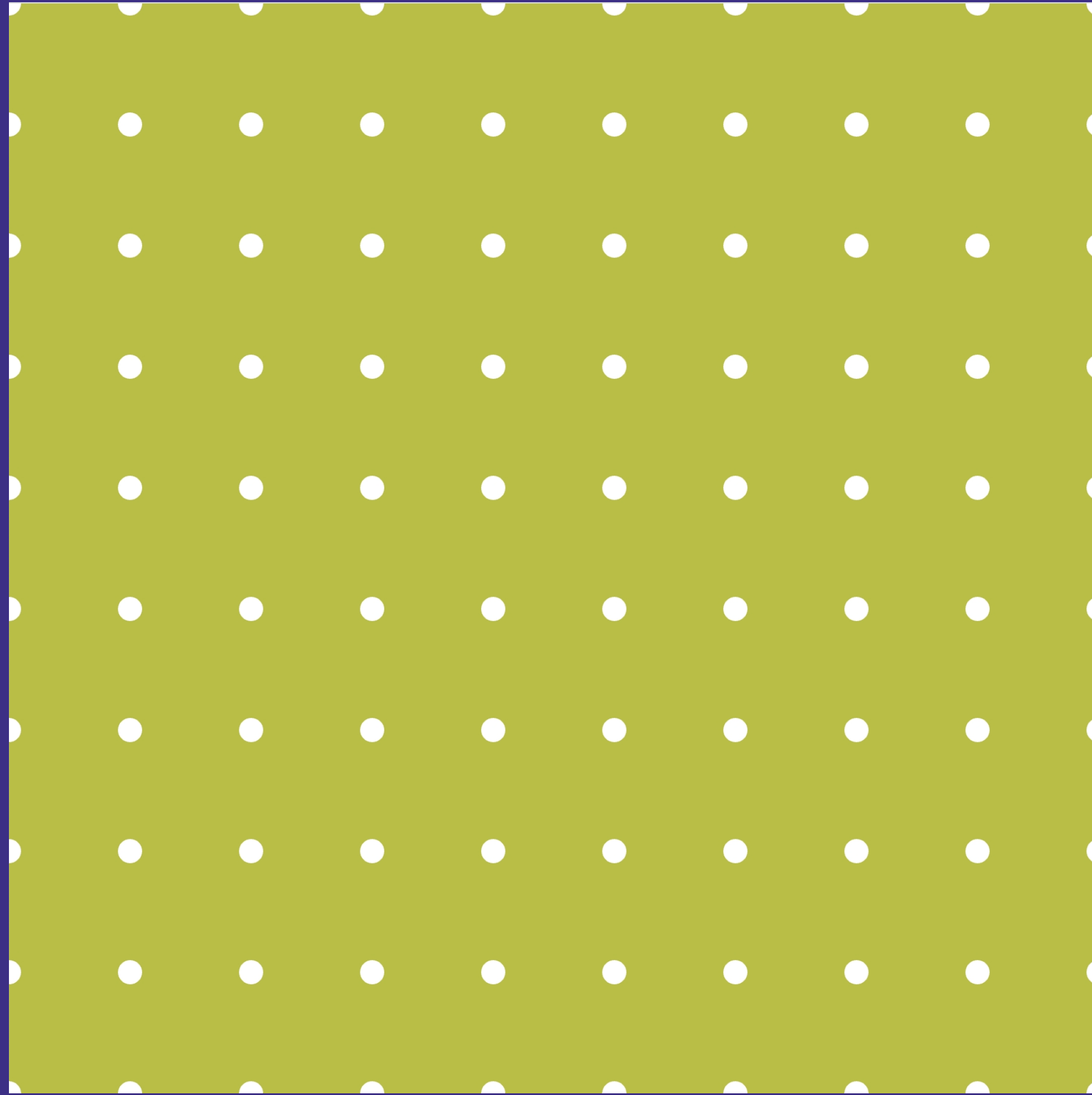
Session 4

Nahuel Gerth,
London College of
Contemporary Arts,
14.11.2023.

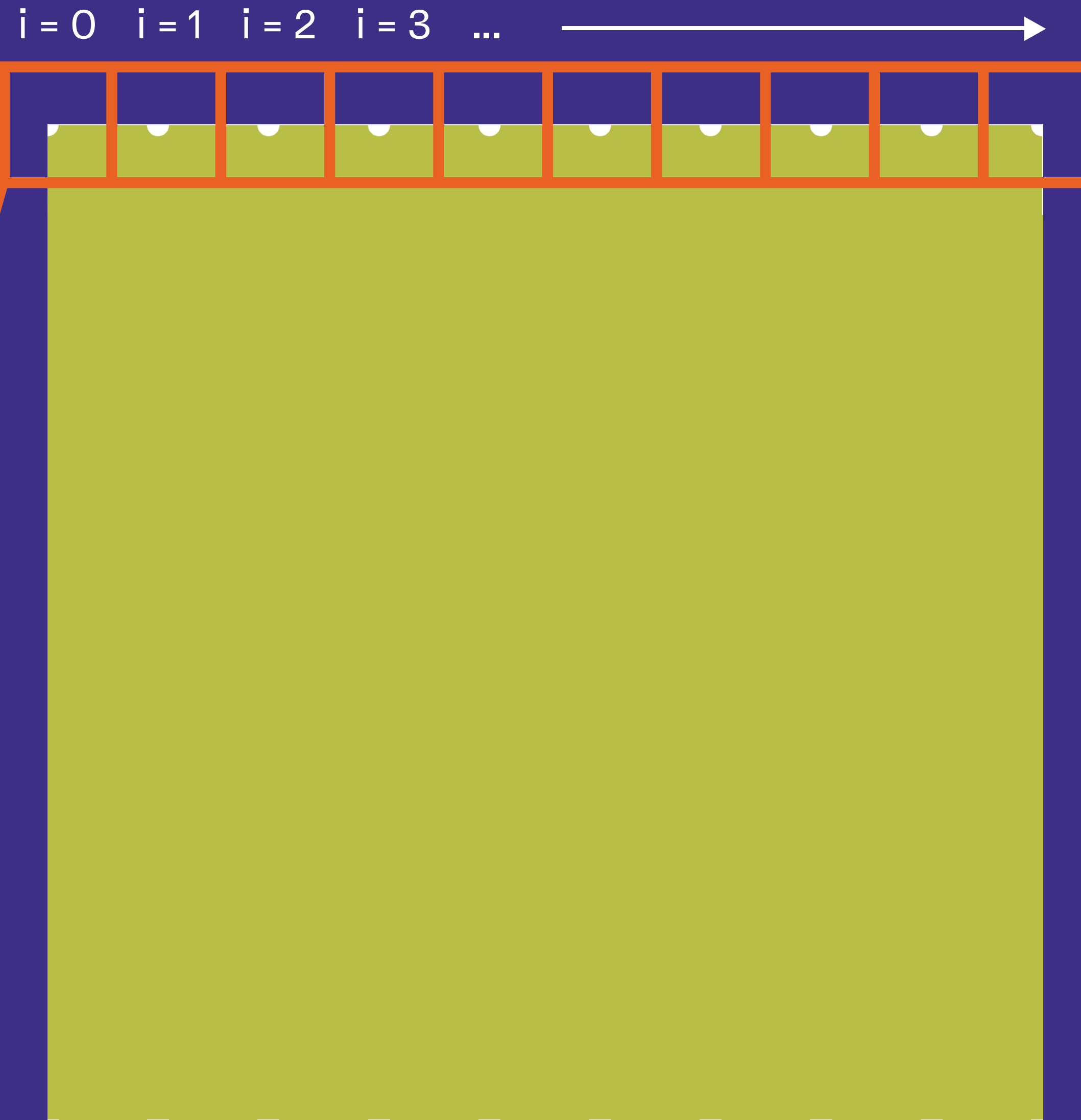
- grid loops
- map
- text
- blend modes
- outlook

nested loops

what is
happening
here?



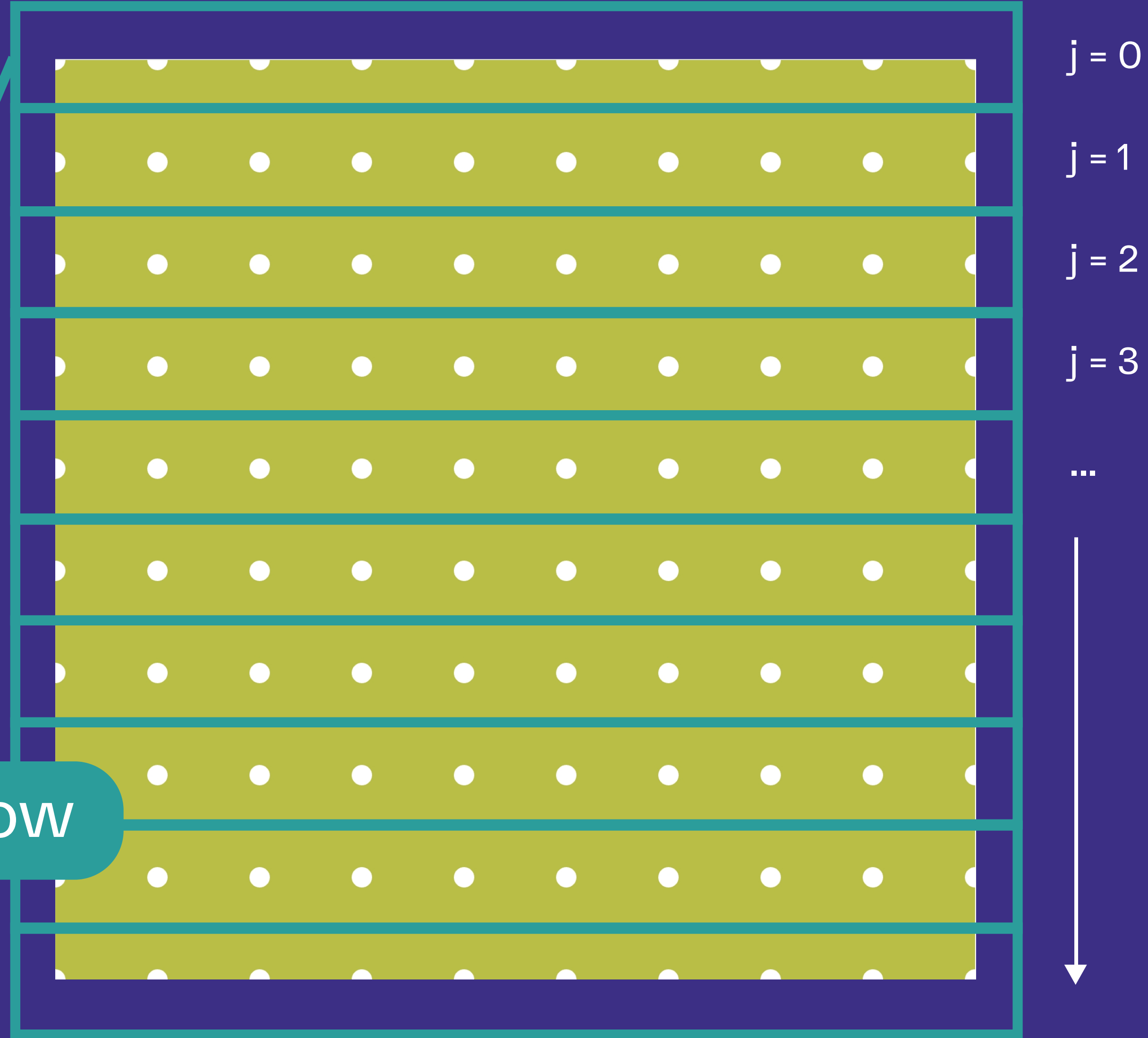
loop 1



1. loop the ellipse

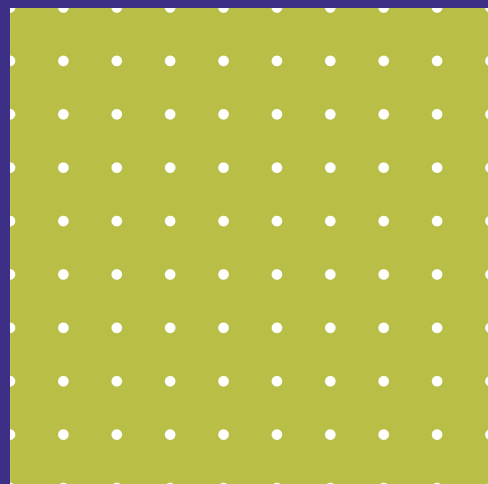
loop 2

2. loop the whole row



nested loops

- we have **two loops nested** inside of each other
- the **inner loop** draws ellipses from left to right
- the **outer loop** repeats the ellipse line from top to bottom



```
let accent = "#B8BE24"; // nice green color

function setup() {
  createCanvas(900, 900);
  noStroke();
}

function draw() {

  background(accent);
  fill("white");

  for (let j = 0; j < 10; j++) {
    for (let i = 0; i < 10; i++) {

      ellipse(i * 100, j * 100, 20, 20);

    }
  }
}
```

nested loops

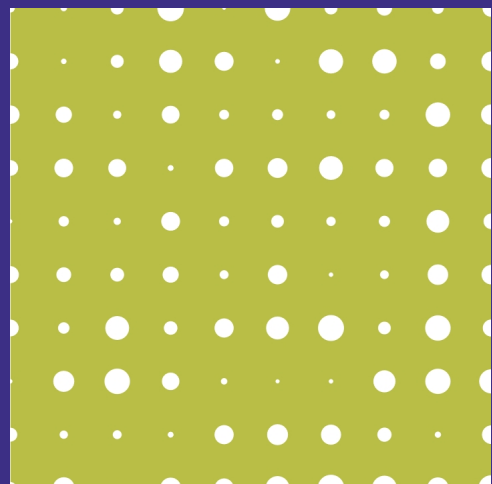
- make each grid element **reactive** to the **mouse** movement
- use **map()** to **narrow down** the mouse values

```
ellipse(i * 100, j * 100, mouseX, mouseY);
```

```
let ellipseWidth;  
ellipseWidth = map(mouseX, 0, width, 0, 20);  
  
ellipse(i * 100, j * 100, ellipseWidth, ellipseWidth);
```

grid with random values

- we create an **array of random values** (same amount as our grid cells)
- we use them to draw **each** grid element **differently**



```
let accent = "#B8BE24"; // nice green color
let sizes = []; // this will be our random sizes

function setup() {
  createCanvas(900, 900);
  noStroke();

  // fill array with random sizes
  for (let i = 0; i < 100; i++) {
    sizes[i] = random(5, 50); // random value between 5 and 50
  }
}

function draw() {

  background(accent);
  fill("white");

  for (let j = 0; j < 10; j++) {
    for (let i = 0; i < 10; i++) {
      let index = j * 10 + i;
      ellipse(i * 100, j * 100, sizes[index], sizes[index]);
    }
  }
}
```


map function

```
map(mouseX, 0, width, 0, 100);
```

map function

```
map(mouseX, 0, width, 0, 100);
```

take this

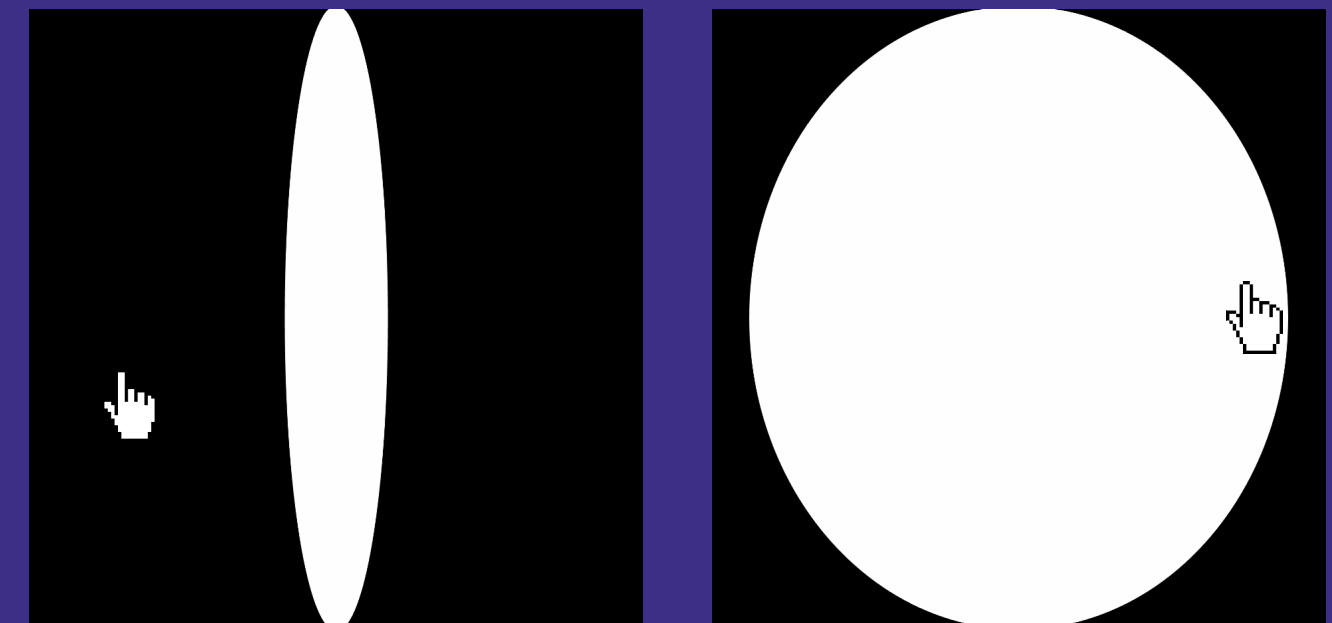
now recalculate the
values between this range

to this range

map function + mouse

```
1  let boxWidth;
2
3  function setup() {
4    |  createCanvas(900, 900);
5  }
6
7  function draw() {
8
9    |  boxWidth = map(mouseX, 0, width, 0, 500);
10
11   |  background("black");
12   |  ellipse(width / 2, height / 2, boxWidth, 400);
13   |  }
14
```

remap the mouse values
to ones that we like more



text

.Aa

text("Aa", x, y);

text styling

→ size

```
textSize(20);
```

→ alignment (x, y)

```
textAlign(CENTER, CENTER);
textAlign(LEFT, TOP);
textAlign(RIGHT, BOTTOM);
```

→ fill

```
noFill();
fill("#FFBA00");
```

→ stroke

```
noStroke();
noFill();
stroke("#FFBA00");
strokeWeight(20);
```

load fonts

- **save** your font to your sketch **folder**
- create a **variable** to hold the font
- **preload()**
load the font from your folder
- **textFont()**
set the font before drawing text

```
let myFont;

function preload(){
  myFont = loadFont("my-nice-font.ttf");
}

function setup() {
  createCanvas(900, 900);
  textFont(myFont);
}

function draw() {
  background(255);

  text("Meat text in nice font", 100, 0, 900, 900);
}
```

custom cursor

- **hide** the default **cursor**
- draw an **ellipse** at **mouse position**
- draw **text** at mouse position and show **mouse values**



```
let accent = "#B8BE24"; // nice green color

function setup() {
  createCanvas(900, 900);
  noStroke();
  noCursor(); // hide the cursor
}

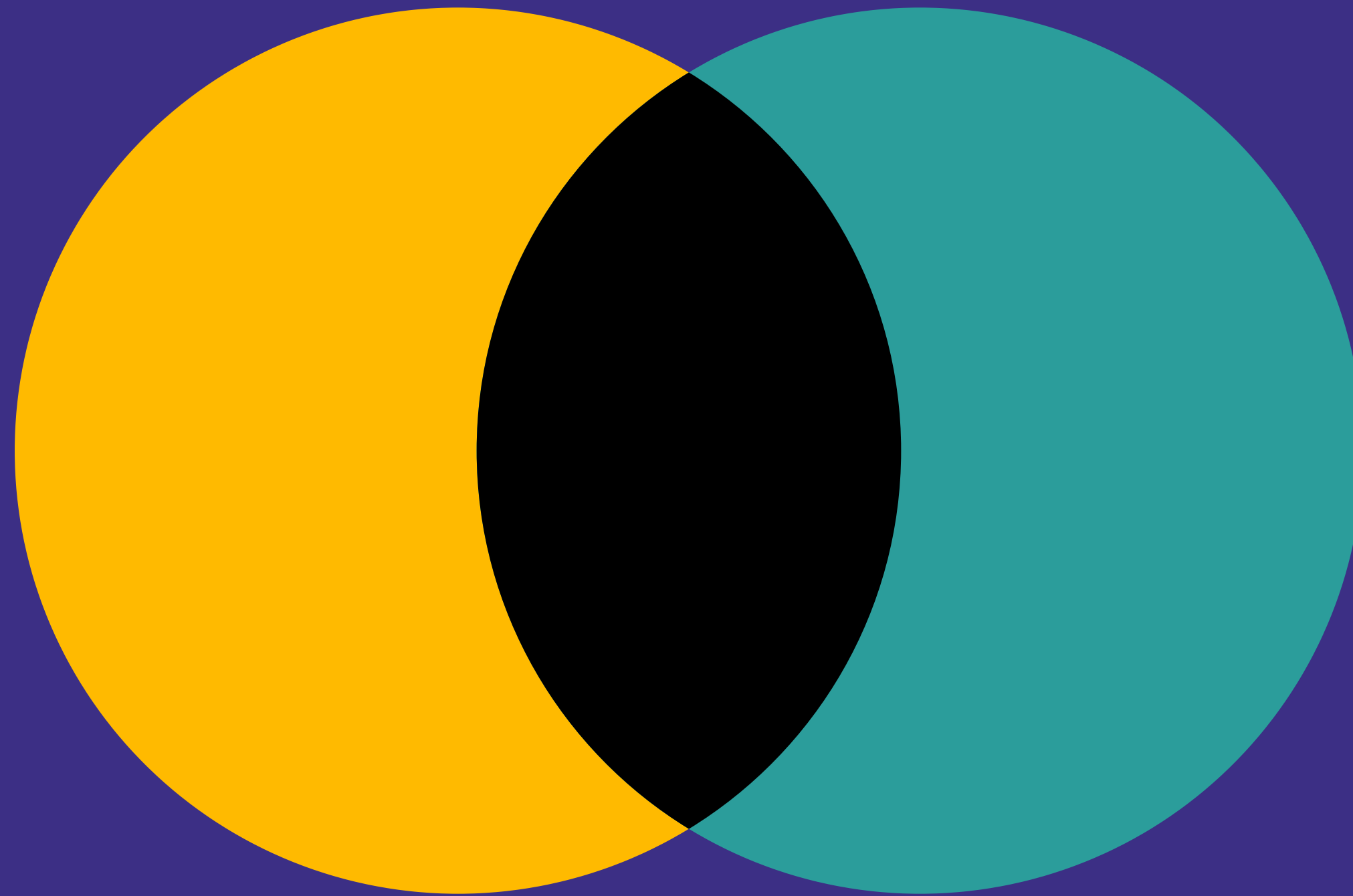
function draw() {

  background(accent);

  // draw cursor
  fill("white");
  ellipse(mouseX, mouseY, 50, 50);

  // draw text
  fill("black");
  textSize(50);
  textAlign(LEFT, CENTER);
  text("x(" + mouseX + ") y(" + mouseY + ")", mouseX + 40, mouseY);
}
```

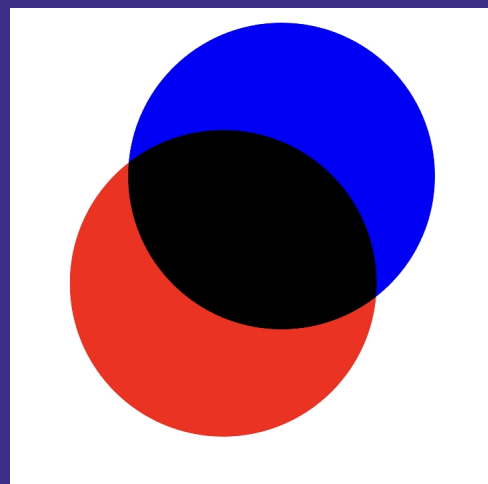
blend mode



```
blendMode(MULTIPLY);
```

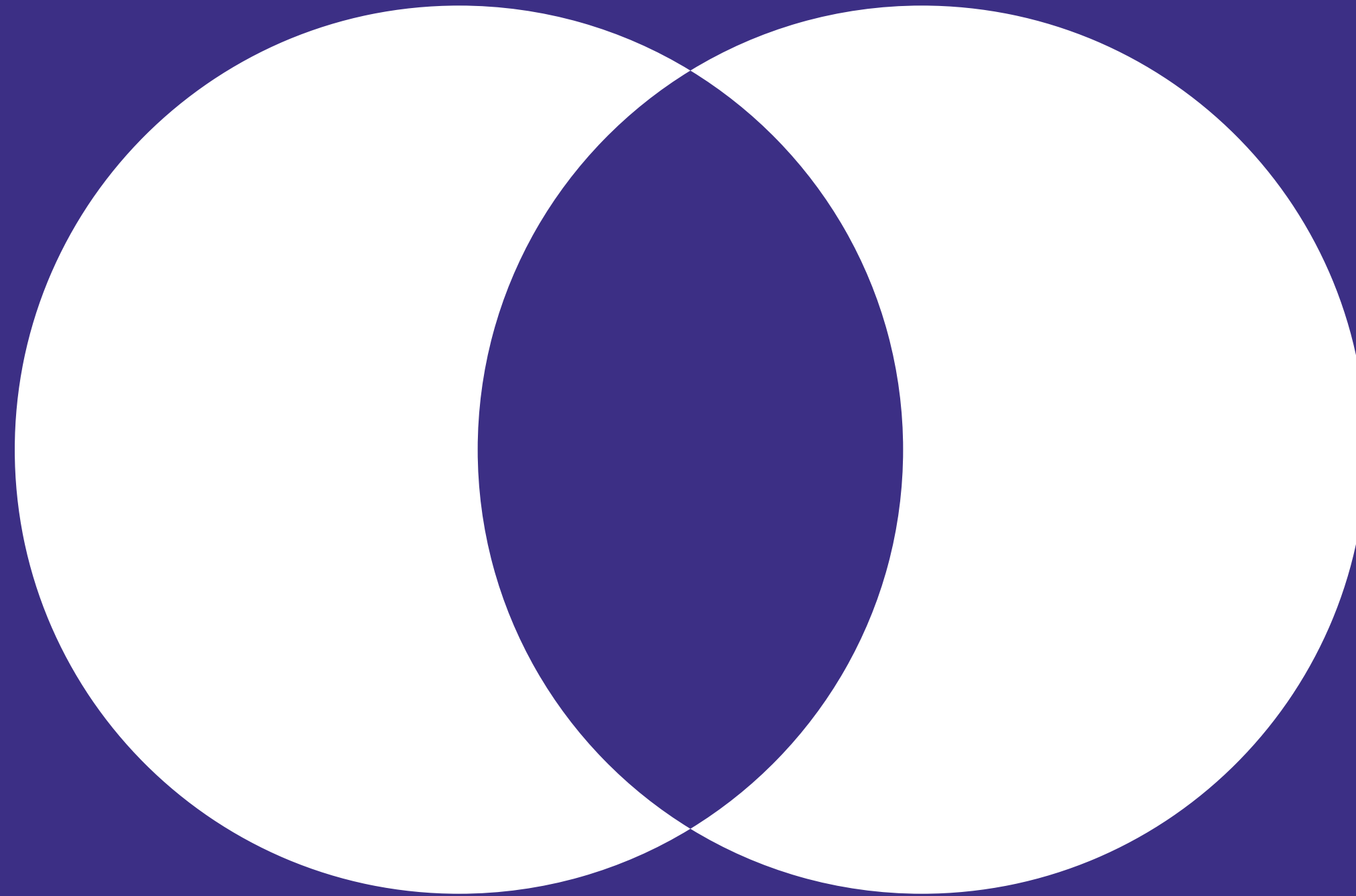

multiply

- draw background with **default blend** mode
- draw first ellipse
- draw second ellipse with **multiply blend** mode



```
function setup() {  
  createCanvas(900, 900);  
  noStroke();  
}  
  
function draw() {  
  
  blendMode(BLEND);  
  background("white");  
  
  fill("red");  
  ellipse(width / 2, height / 2, 500, 500);  
  
  blendMode(MULTIPLY);  
  fill("blue");  
  ellipse(mouseX, mouseY, 500, 500);  
}
```

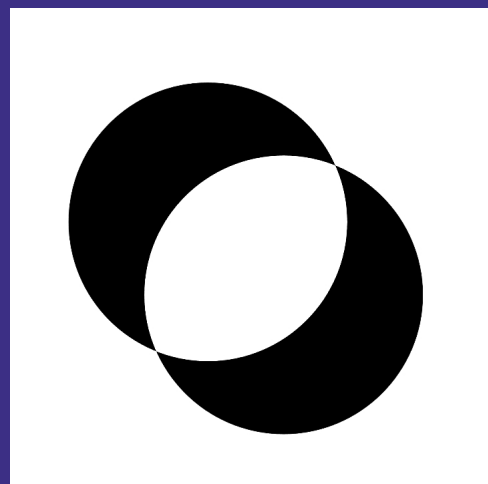
blend mode



```
blendMode(DIFFERENCE);
```

subtract

- draw background with **default blend** mode
- set blend mode to **difference**
- draw the two **ellipses**



```
function setup() {
  createCanvas(900, 900);
  noStroke();
}

function draw() {

  blendMode(BLEND);
  background("white");

  blendMode(DIFFERENCE);
  fill("white");
  ellipse(width / 2, height / 2, 500, 500);
  ellipse(mouseX, mouseY, 500, 500);
}
```

Resources & inspiration

Zach Lieberman

[instagram.com/zach.lieberman](https://www.instagram.com/zach.lieberman)

daily code experiments

Vera van de Seyp

[instagram.com/veravandeseyp](https://www.instagram.com/veravandeseyp)

crazy type experiments

Tim Rodenbröker

timrodenbroeker.de

onboarding courses

The Coding Train

youtube.com/@TheCodingTrain

endless tutorials

Studio Dumber

studiodumber.com

beautiful motion design

kollektiv_generativ

[instagram.com/kollektiv_generativ](https://www.instagram.com/kollektiv_generativ)

creative coding research & experiments

by Katja Rempel & Nahuel Gerth