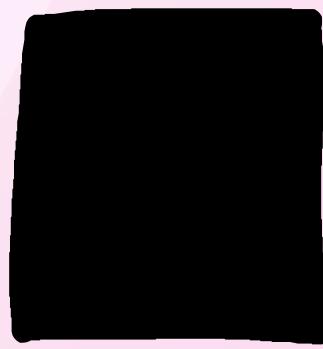


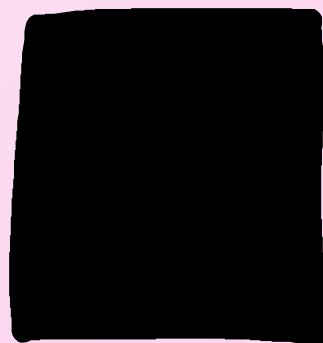
MediaArt Programming

20210727

本日のテーマ



反復と変数



乱数

>

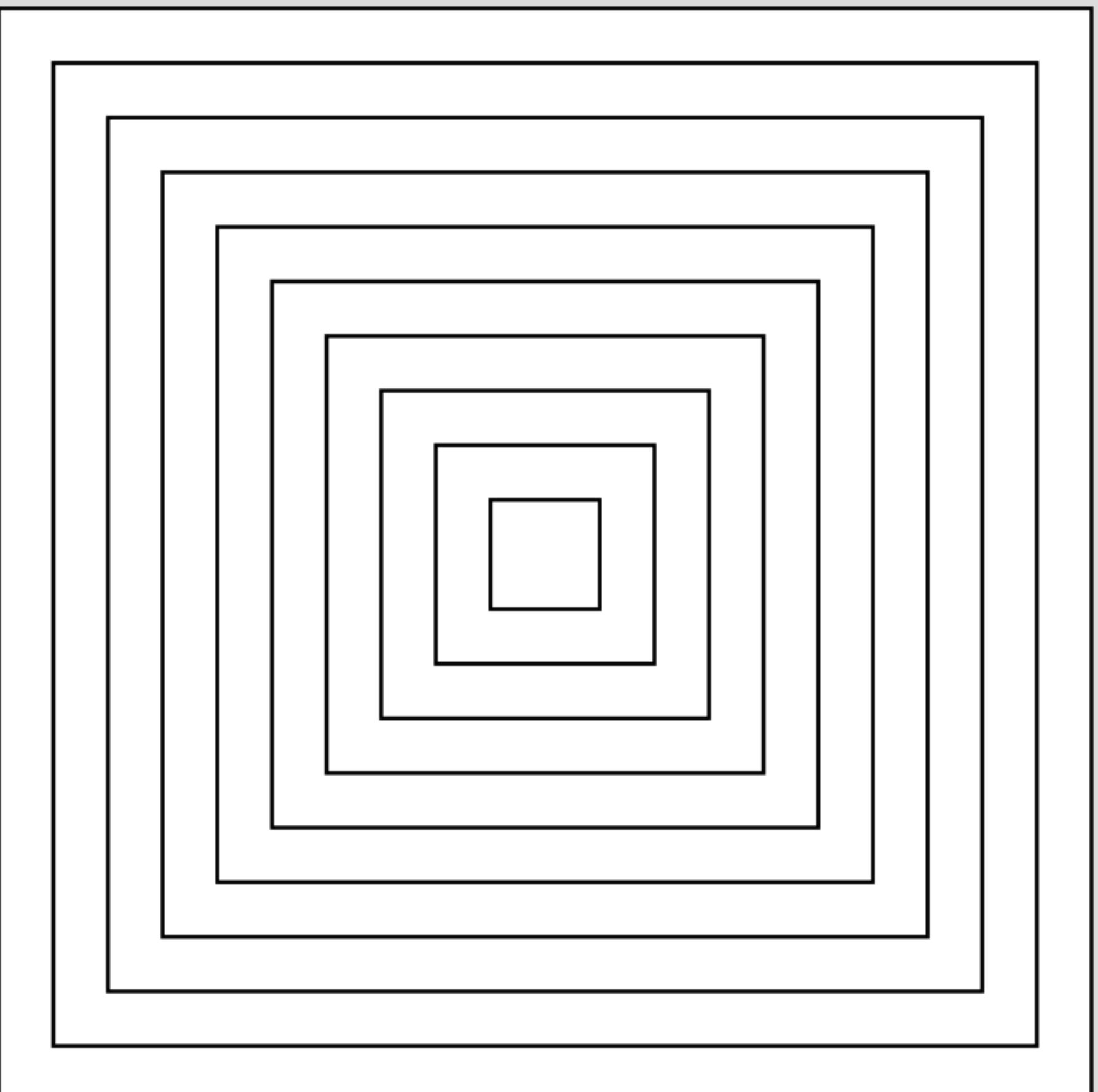
sketch.js*

Preview

```
1 function setup() {  
2   createCanvas(600,600);  
3   background(0,0,0)  
4   fill(255,255,255)  
5   rect(0,0,300,600)  
6  
7   fill(0,0,0)  
8   ellipse(150,150,150)  
9   fill(255,255,255)  
10  ellipse(150,150,125)  
11  fill(0,0,0)  
12  ellipse(150,150,100)  
13  rect(135,150,30,300)  
14  rect(135,400,80,20)  
15  rect(135,370,100,20)  
16  
17  fill(255,255,255)  
18  ellipse(450,250,150)  
19  
20  fill(0,0,0)  
21  ellipse(450,250,125)  
22  
23  
24  fill(255,255,255)  
25  rect(350,250,200,150)  
26  
27  fill(0,0,0)  
28  ellipse(450,300,60)  
29  triangle(450,300,410,380,490,380)  
30  
31  
32  
33 }  
34
```



Preview

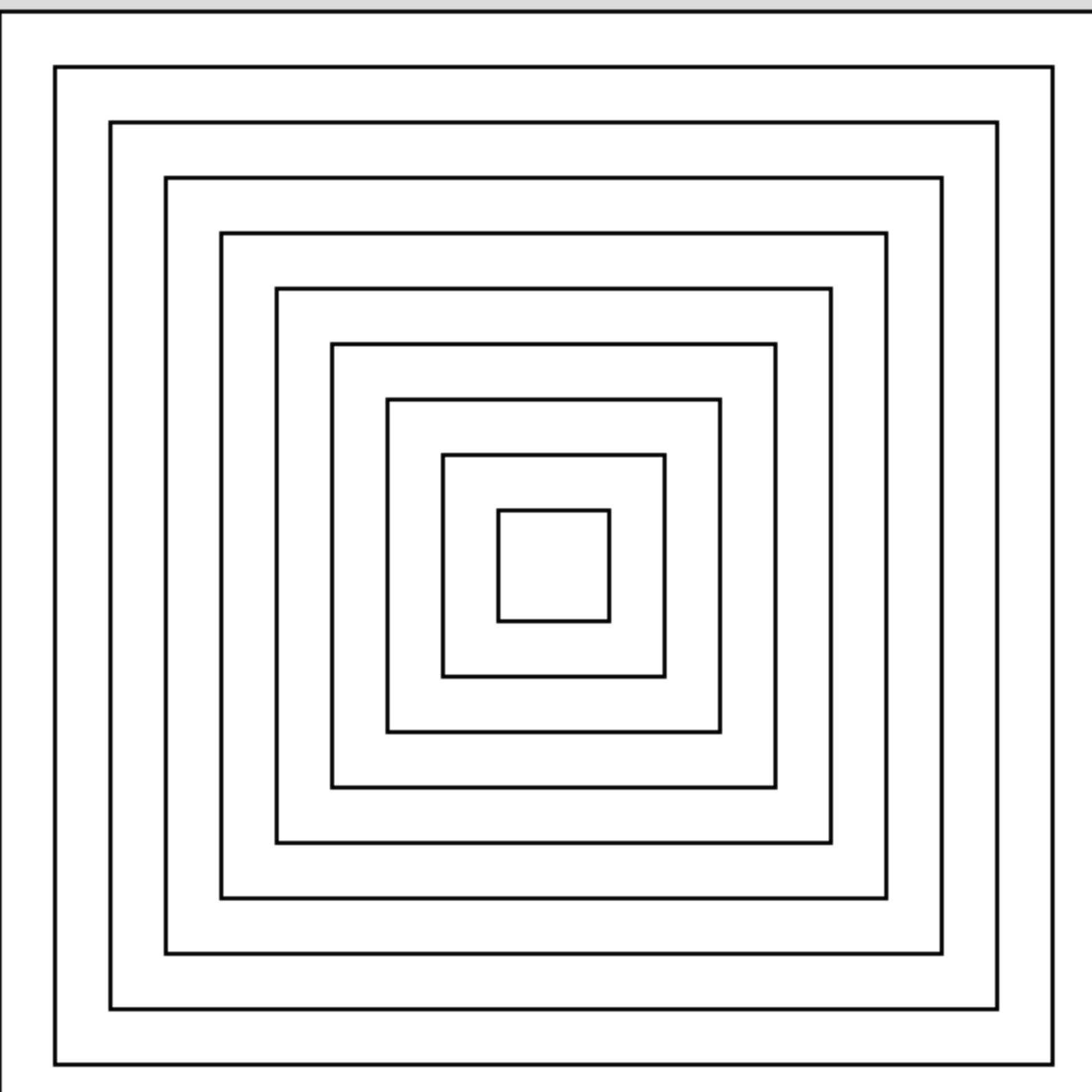


>

sketch.js

```
1  function setup() {  
2      createCanvas(50, 50);  
3  }  
4  
5  function draw() {  
6      background(220);  
7  
8      rect(1,1,48)  
9      rect(3.5,3.5,43)  
10     rect(6,6,38)  
11     rect(8.5,8.5,33)  
12     rect(11,11,28)  
13     rect(13.5,13.5,23)  
14     rect(16,16,18)  
15     rect(18.5,18.5,13)  
16     rect(21.5,21.5,8)  
17     rect(24,24,3)  
18  
19 }
```

Preview



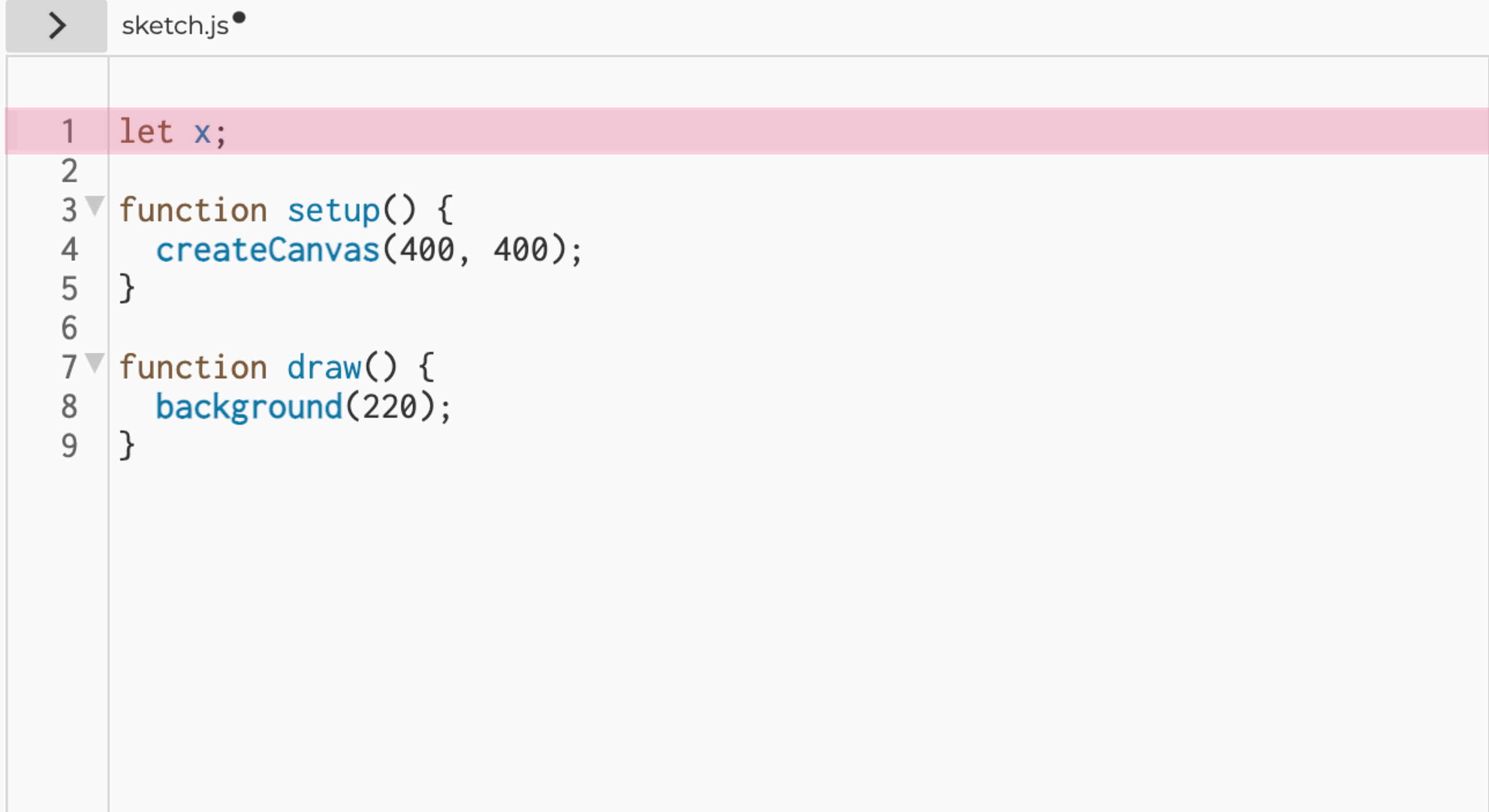
变量

1. 宣言(Declare)
2. 初始化(Initialize)
3. 参照(Reference)

JavaScript

① 宣言(Declare)

```
let ○○○;
```



The screenshot shows a code editor window with a file named "sketch.js". The code is as follows:

```
> sketch.js•  
1 let x;  
2  
3▼ function setup() {  
4   createCanvas(400, 400);  
5 }  
6  
7▼ function draw() {  
8   background(220);  
9 }
```

The first line, "let x;", is highlighted with a pink background. The code uses indentation to show the structure of functions: "setup" and "draw". The "background" function call is also highlighted.

变量

② 初期化 (Initialize)

```
let ○○○;
```

```
> sketch.js•  
1 let x;  
2  
3▼ function setup() {  
4   createCanvas(400, 400);  
5 }  
6  
7▼ function draw() {  
8   background(220);  
9  
10  x = 50;  
11  
12 }
```

变量

③ 参照(Reference)

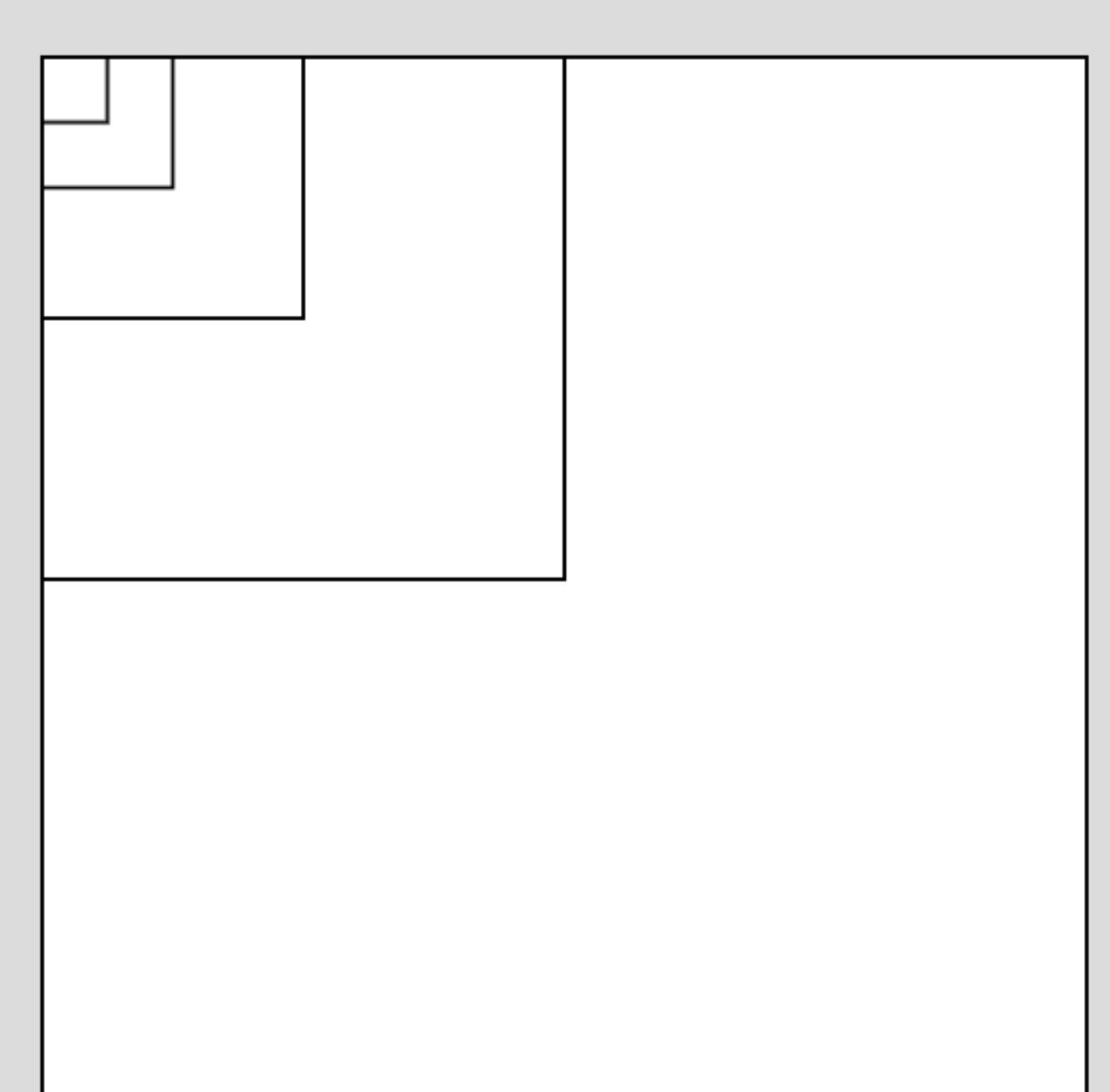
```
let ○○○;
```

```
> sketch.js•  
1 let x;  
2  
3▼ function setup() {  
4   createCanvas(400, 400);  
5 }  
6  
7▼ function draw() {  
8   background(220);  
9  
10  x = 50;  
11  rect(x, 100, 150);  
12  
13 }
```

反復

> sketch.js • Preview

```
1 let diameter;
2
3 ▼ function setup() {
4   createCanvas(400, 400);
5 }
6
7 ▼ function draw() {
8   background(220);
9
10  diameter = 300;
11  rect(50, 50, diameter);
12
13  diameter = diameter/2;
14  rect(50, 50, diameter);
15
16  diameter = diameter/2;
17  rect(50, 50, diameter);
18
19  diameter = diameter/2;
20  rect(50, 50, diameter);
21
22  diameter = diameter/2;
23  rect(50, 50, diameter);
24
25 }
```

The preview window displays a fractal tree structure. It starts with a single vertical line segment of length 300. This segment is divided into three smaller segments: one vertical segment of length 150 and two diagonal line segments of length 50 each, forming a T-shape. This process repeats for each of the three new segments, creating a total of nine segments per level. The tree has four levels of recursion, resulting in a complex, branching pattern that occupies most of the 400x400 canvas area.

反復

```
for( 初期化式 ; 繼続条件式 ; 再初期化式 ){ ○○○ } ;
```

- ◎ 初期化式：初期化の際の条件式
- ◎ 繼続条件式：繰り返しを継続する条件式
- ◎ 再初期化式：繰り返されるたびに実行される式

反復

for(初期化式 ; 繼續條件式 ; 再初期化式){ ○○○ } ;

```
> sketch.js•  
1 let diameter = 300;  
2  
3▼ function setup() {  
4   createCanvas(400, 400);  
5   background(220);  
6  
7▼   for(let i = 0; i < 5; i++){  
8  
9  
10  }  
11  
12 }  
13  
14▼ function draw() {  
15  
16 }
```

反復

```
for( 初期化式 ; 繼續條件式 ; 再初期化式 ){ ○○○ } ;
```

```
> sketch.js•  
1 let diameter = 300;  
2  
3▼ function setup() {  
4   createCanvas(400, 400);  
5   background(220);  
6  
7▼   for(let i = 0; i < 5; i++){  
8     rect(50, 50, diameter);  
9     diameter = diameter / 2;  
10    }  
11  
12  }  
13  
14▼ function draw() {  
15  
16  }
```

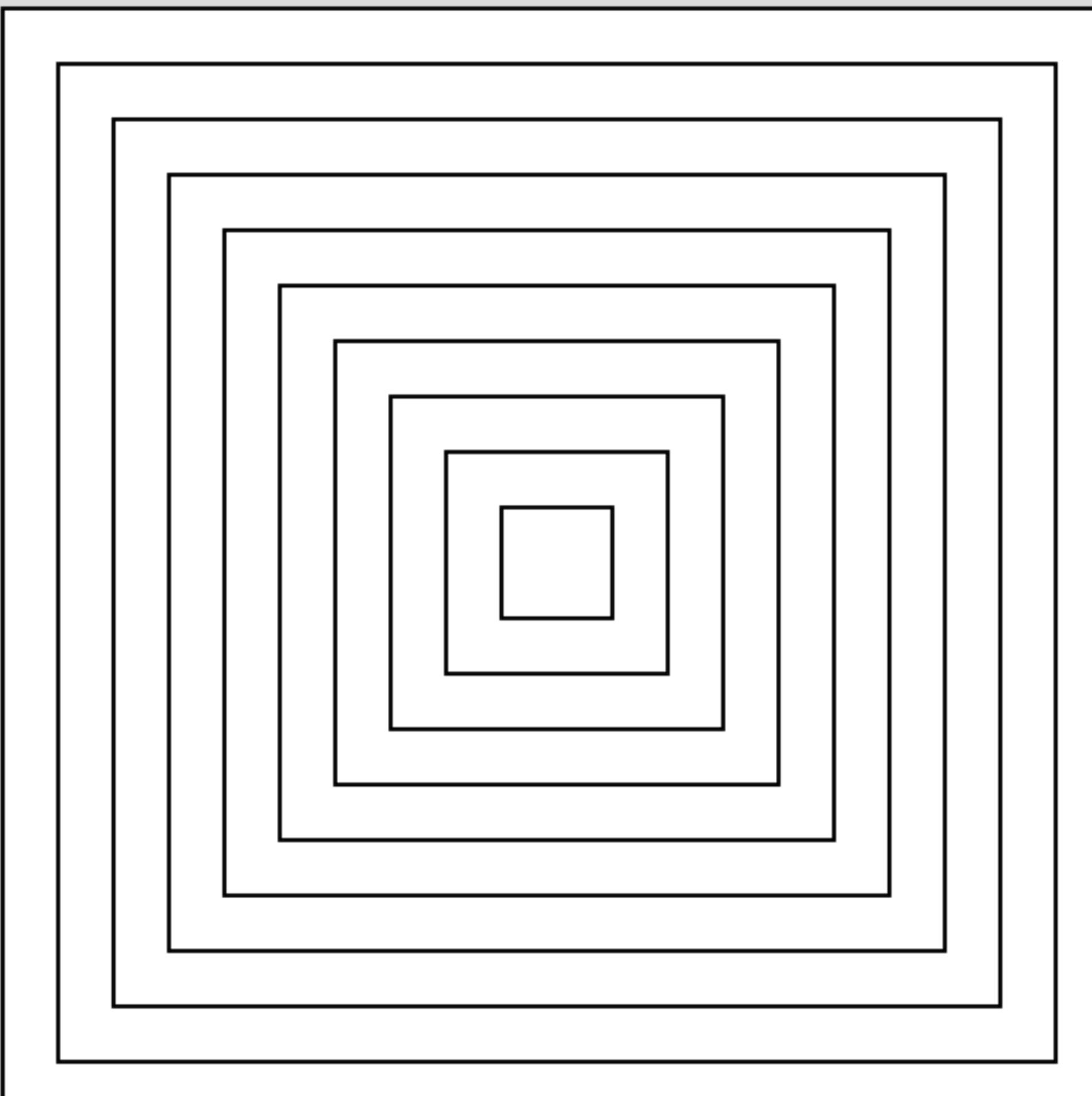
練習問題

Canvas(400, 400)

rect(50, 50, 300)

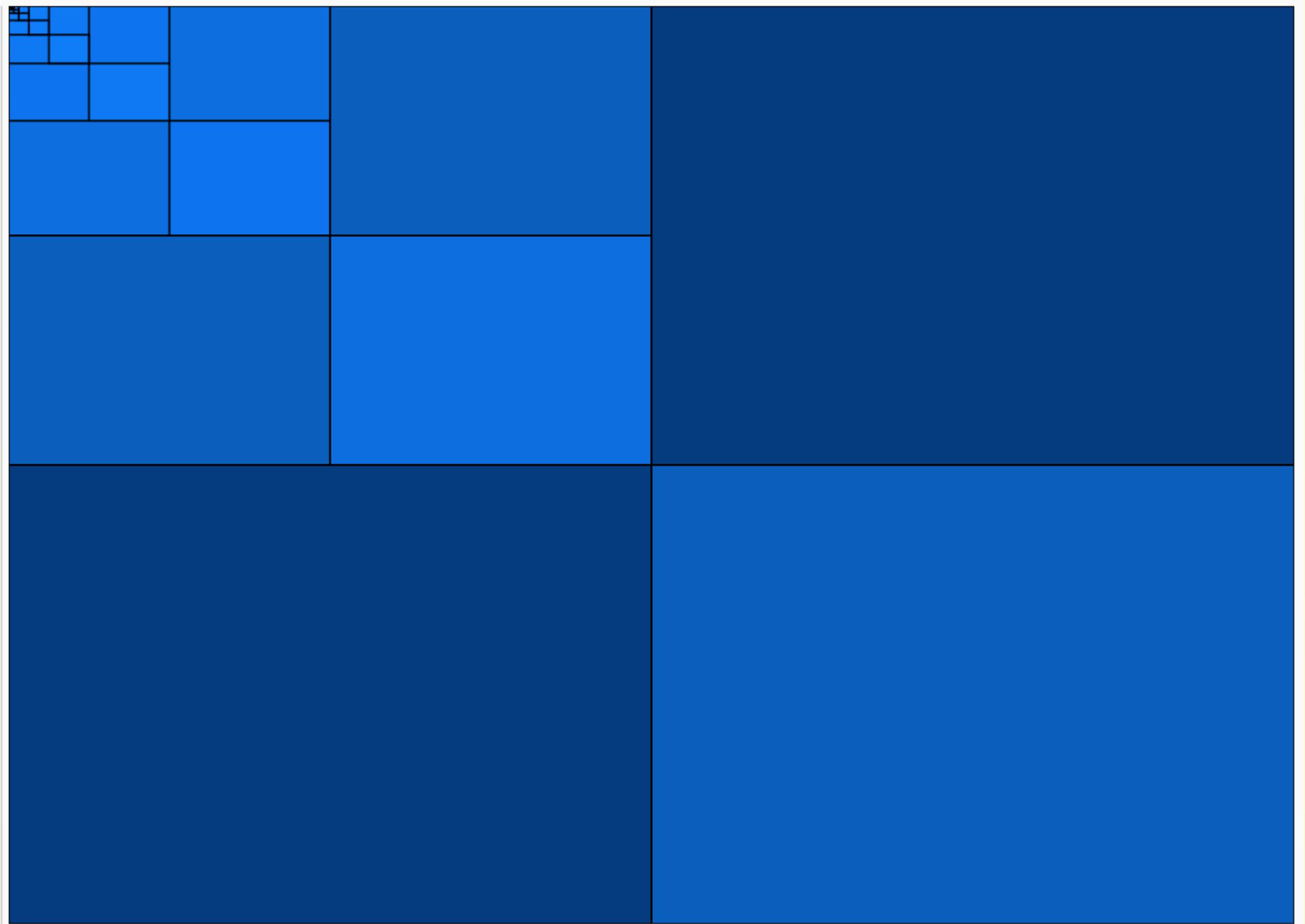
ループ数：10

Preview

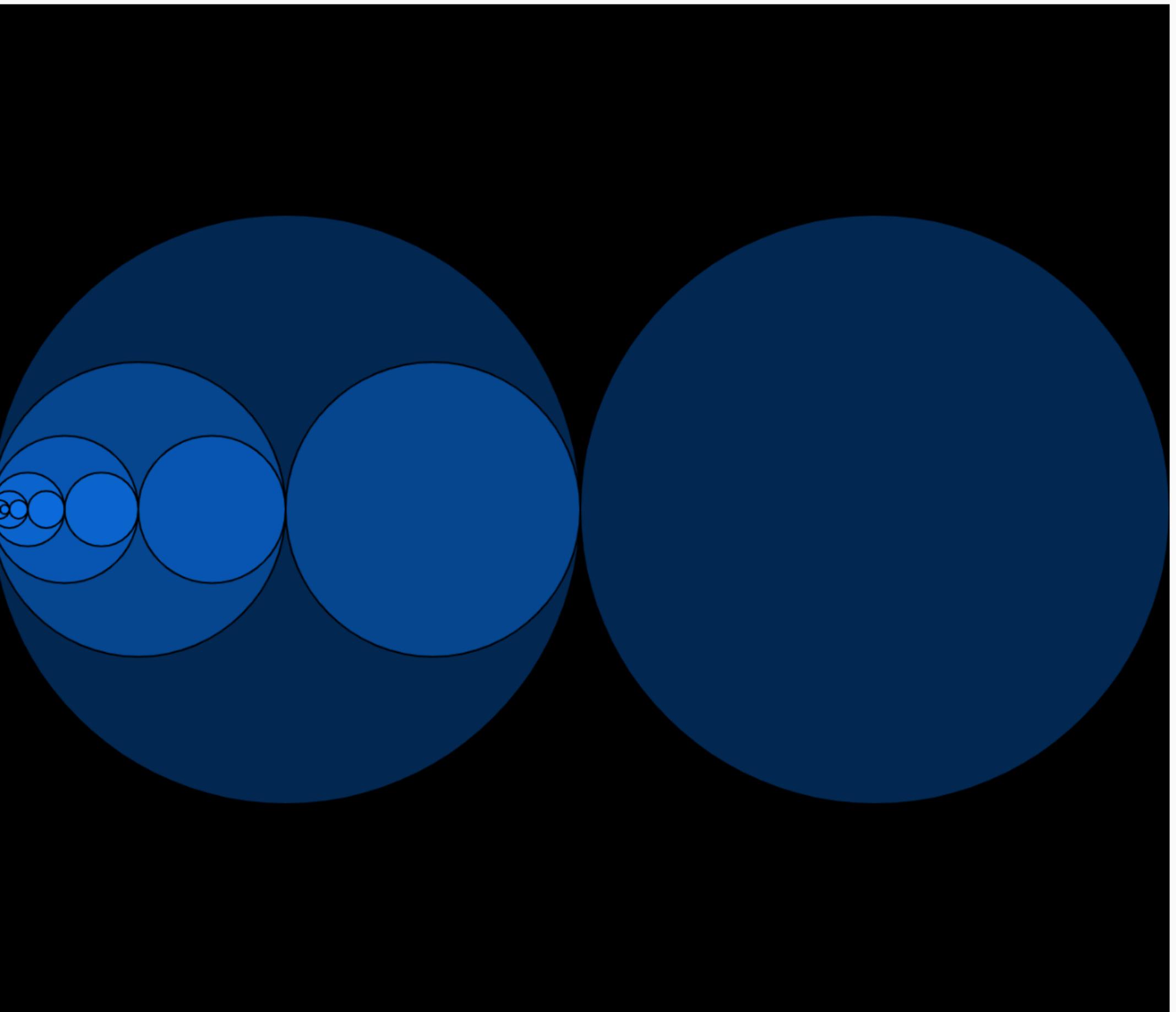


練習向題

Preview



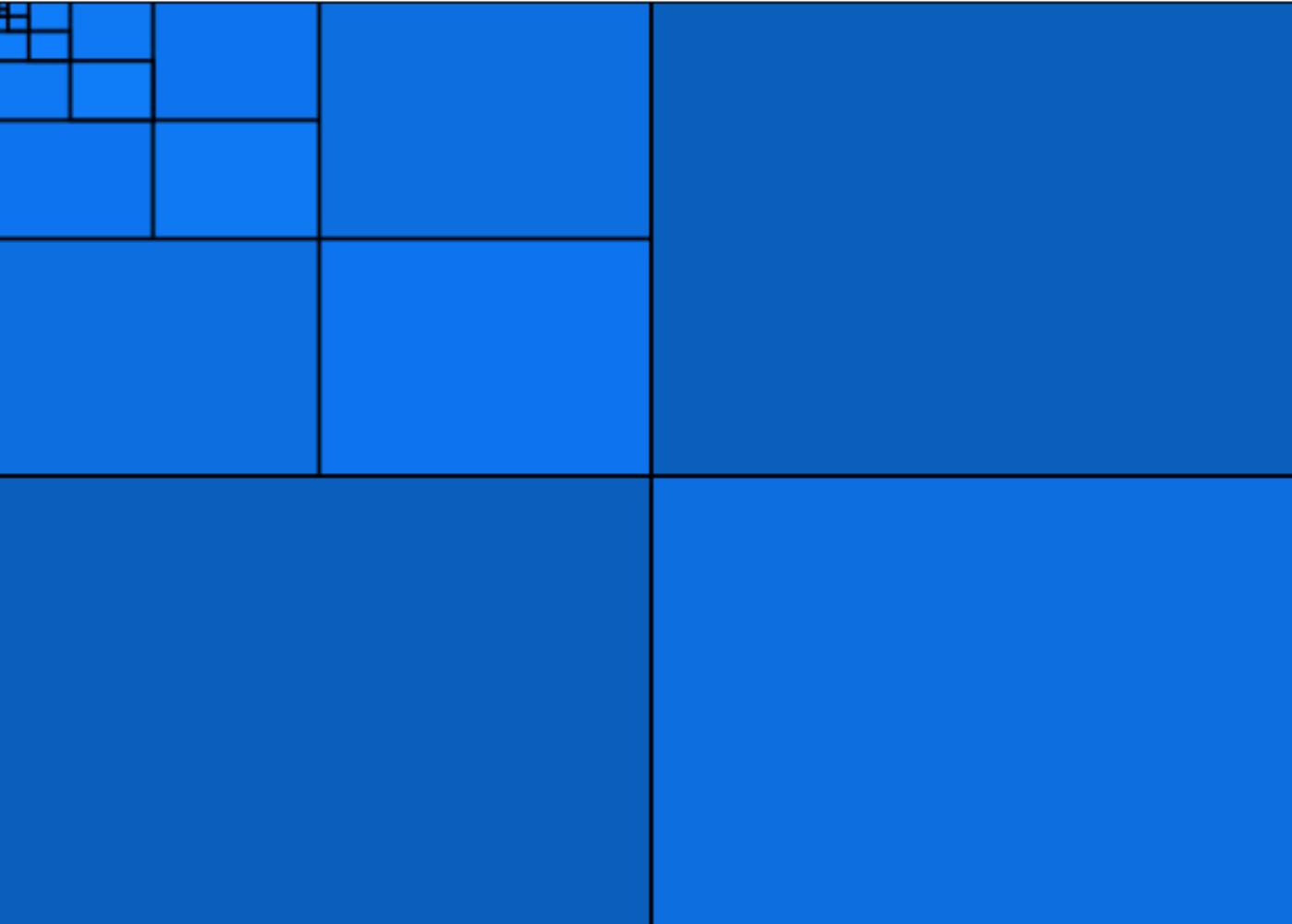
Preview



練習問題

> sketch.js Saved: about 2 months ago Preview

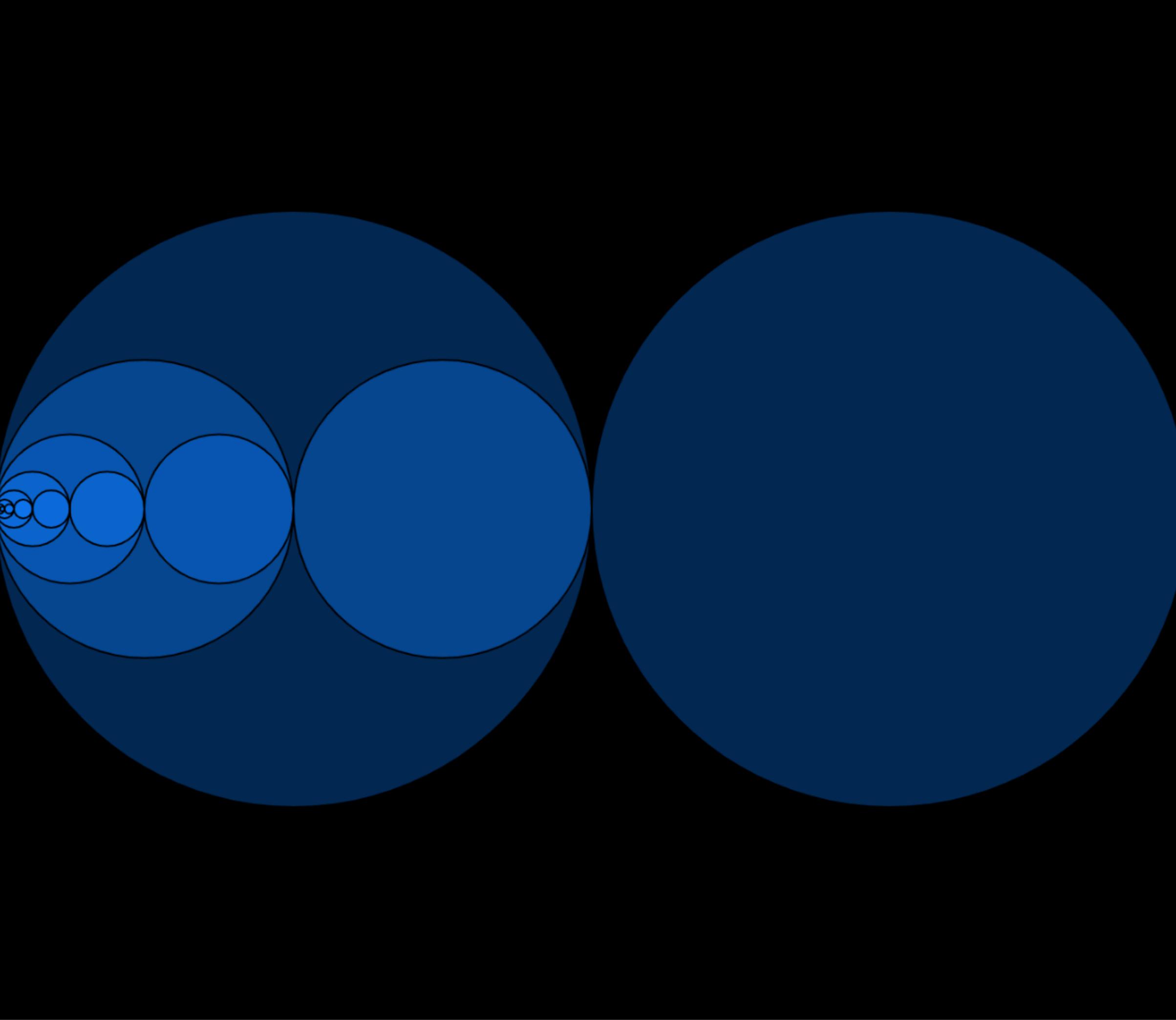
```
1 function setup() {
2   createCanvas(700, 500);
3   stroke(0);
4   fill(0, 127, 255, 127);
5   background(0);
6
7   let x = 0;
8   let y = 0;
9   let w = width;
10  let h = height;
11
12  for(let i = 0; i<10; i++){
13    rect(x, y, w, h);
14    w = w/2;
15    h = h/2;
16    rect(w, h, w, h);
17  }
18}
19
```



練習問題

> sketch.js Saved: about 2 months ago Preview

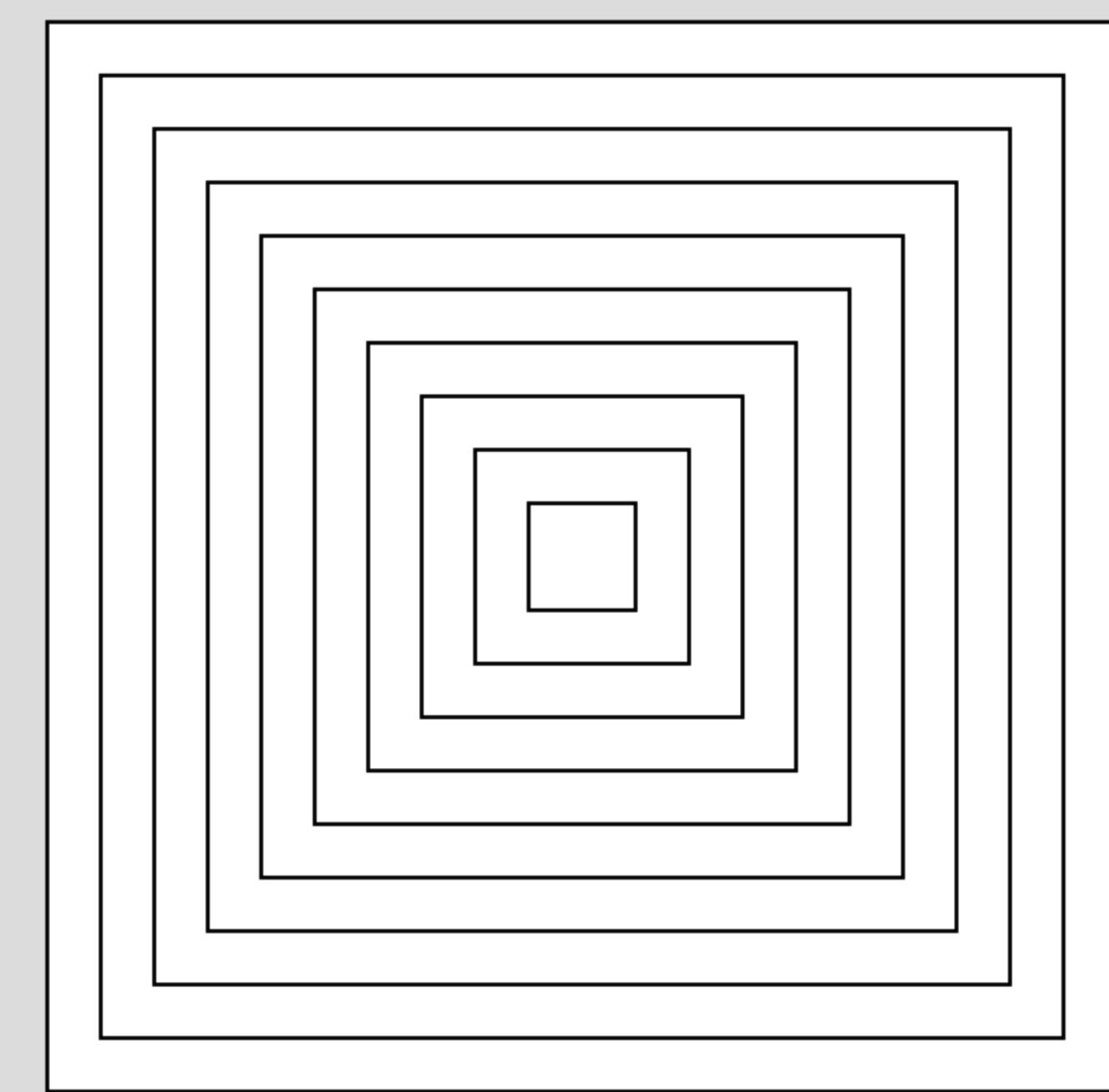
```
1 function setup() {
2   createCanvas(700, 600);
3   stroke(0);
4   fill(0, 127, 255, 85);
5   background(0);
6
7   let w = width;
8   let h = height;
9
10  for(let i = 0; i < 10; i++){
11    circle(w/4, h/2, w/2);
12    circle(w/4*3, h/2, w/2);
13    w = w/2;
14  }
15}
16
```



亂数

random();

Preview



sketch.js •

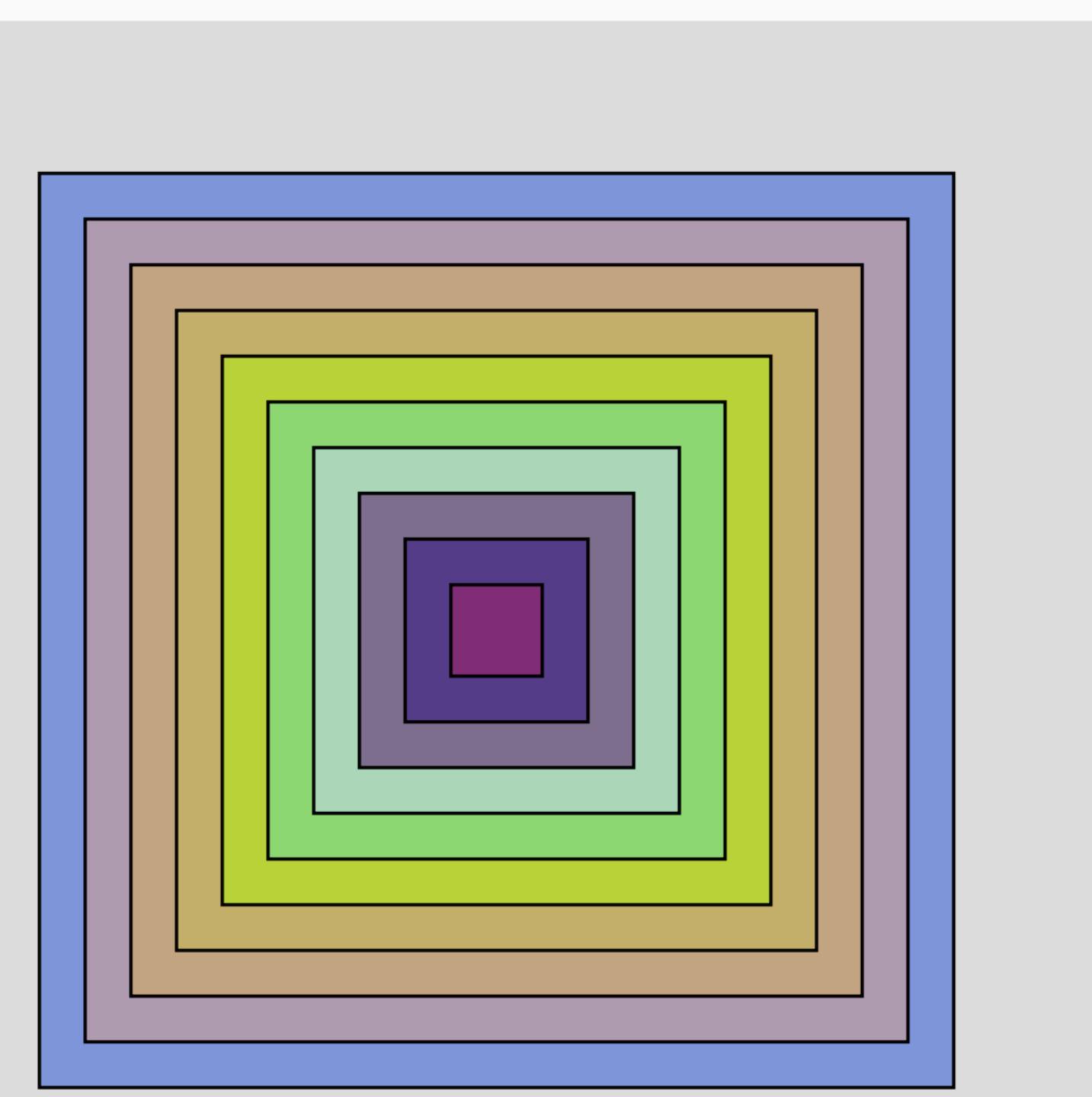
```
1 let x = 50;
2 let y = 50;
3 let diameter = 300;
4
5 function setup() {
6   createCanvas(400, 400);
7   background(220);
8
9 for(let i = 0; i < 10; i++){
10   rect(x, y, diameter);
11   x += 15;
12   y += 15;
13   diameter -= 30;
14 }
15
16 }
```

亂数

random();

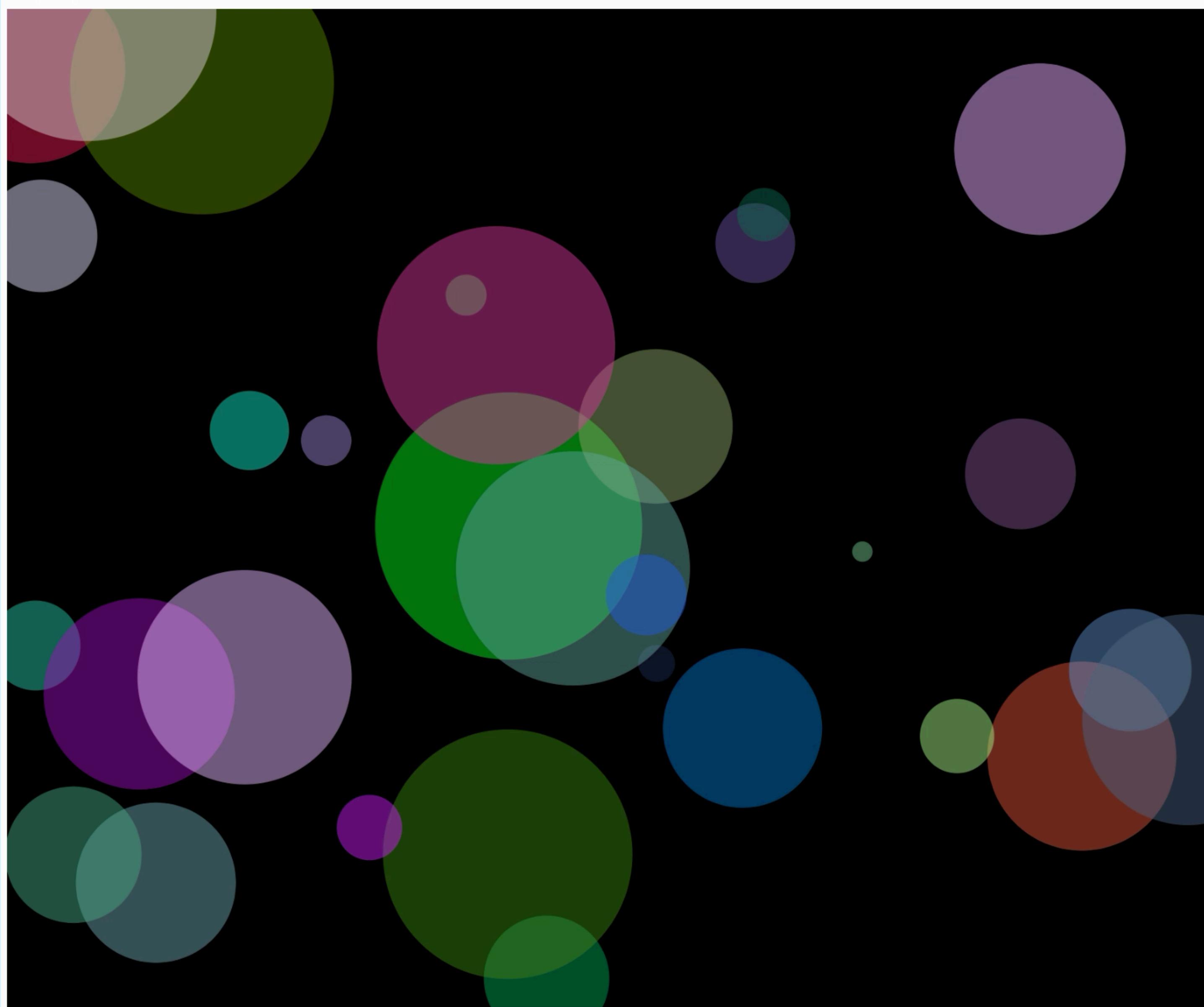
> sketch.js • Saved: 2 minutes ago Preview

```
1 let x = 50;
2 let y = 50;
3 let diameter = 300;
4
5 function setup() {
6   createCanvas(400, 400);
7   background(220);
8
9 for(let i = 0; i < 10; i++){
10   fill(random(0,255), random(0,255), random(0,255), 127);
11   rect(x, y, diameter);
12   x += 15;
13   y += 15;
14   diameter -= 30;
15 }
16
17 }
18
```



亂
朱
散

Preview



亂数

```
random();
```



A screenshot of a code editor window titled "sketch.js". The code is a simple JavaScript function for a Processing sketch. It defines a setup function that creates a canvas, sets the frame rate to 30, and sets the background to black. It also defines a draw function that currently contains no code. The code is numbered from 1 to 9. The setup function is highlighted with a pink background.

```
> sketch.js
Saved: 1 minute ago

1▼ function setup() {
2  createCanvas(windowWidth, windowHeight);
3  frameRate(30);
4  background(0);
5 }
6
7▼ function draw() {
8
9 }
```

亂数

```
random();
```

> sketch.js •

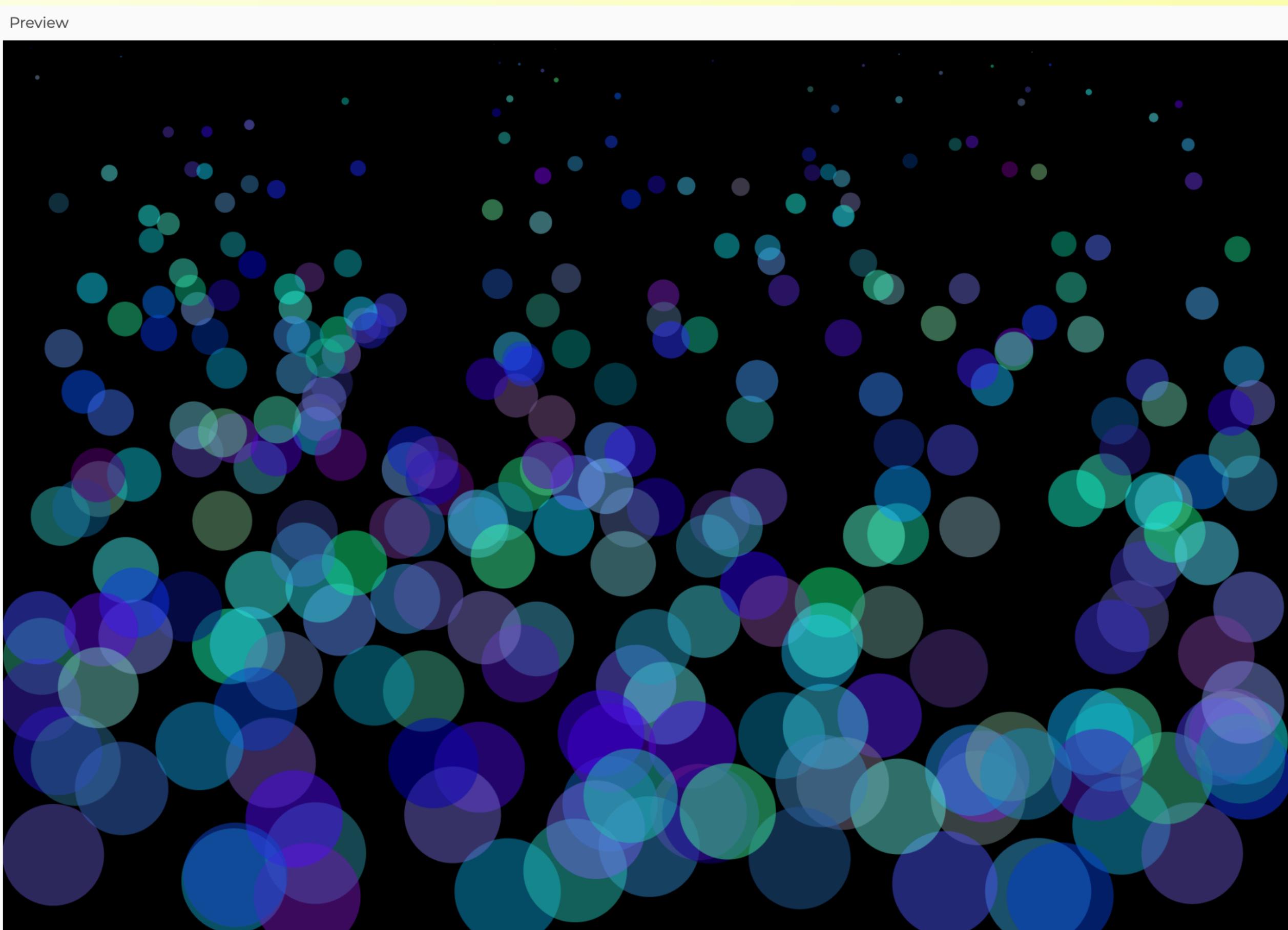
Saved: 4 minutes ago

```
1 function setup() {
2   createCanvas(windowWidth, windowHeight);
3   frameRate(30);
4   background(0);
5 }
6
7 function draw() {
8   noStroke();
9   fill(random(255), random(255), random(255), 127);
10
11  let diameter = random(10, 200);
12
13  ellipse(random(width), random(height), diameter);
14 }
```

自由倉庫ト

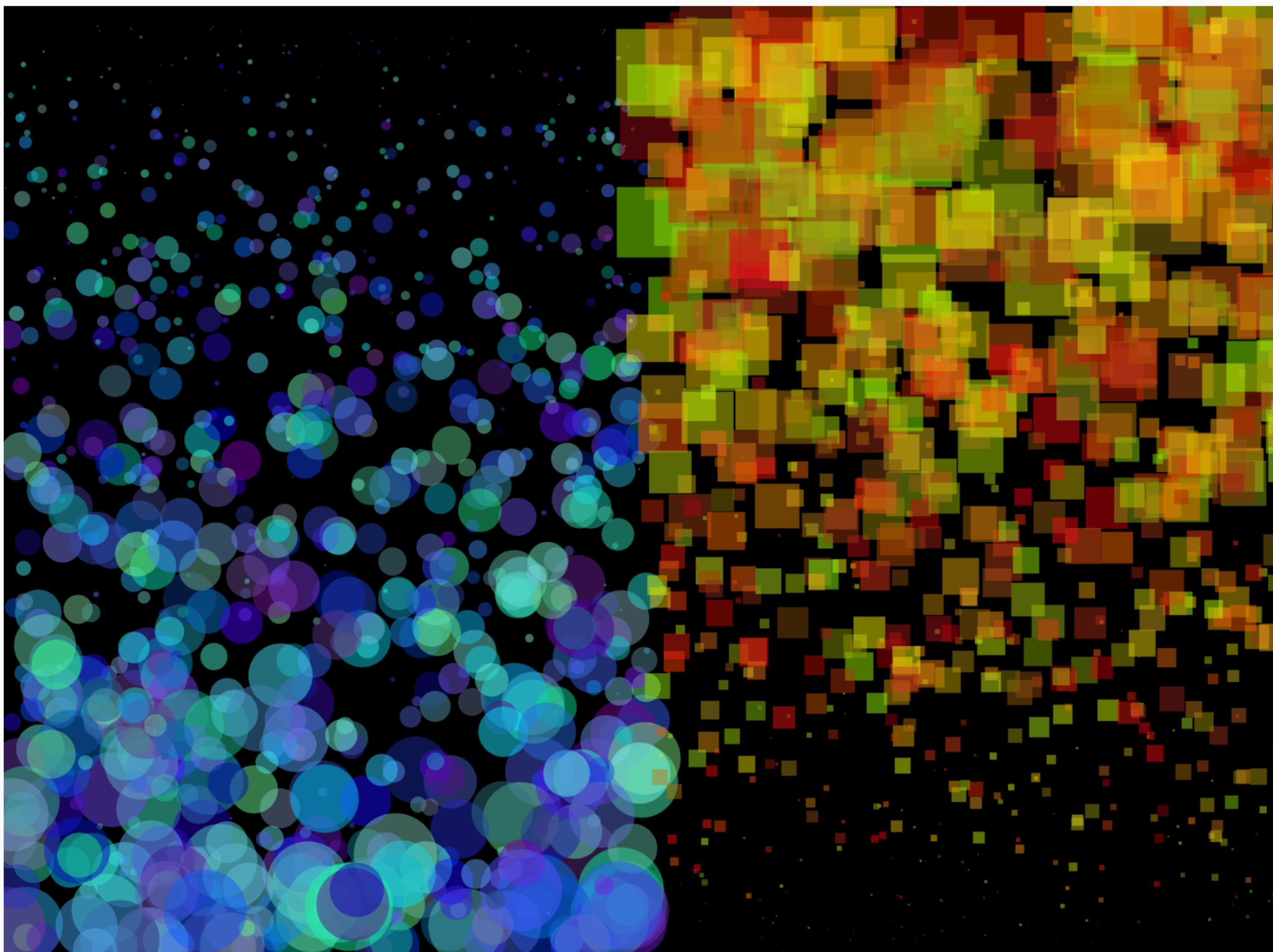
randomを使って、描画に不規則性を持たせよう

例えば....



自由倉庫

Preview



自由倉庫

