

# **Feedback and recursion**



# RECURSION

It recurs.

---

# RECURSION

It recurs.

# RECURSION

It recurs.

It recurs.

# RECURSION

ACCURSI

RECURSION

## **RECURSOS**

SELECT EDITIONS

www.sagepub.com

What exactly is Video Feedback?

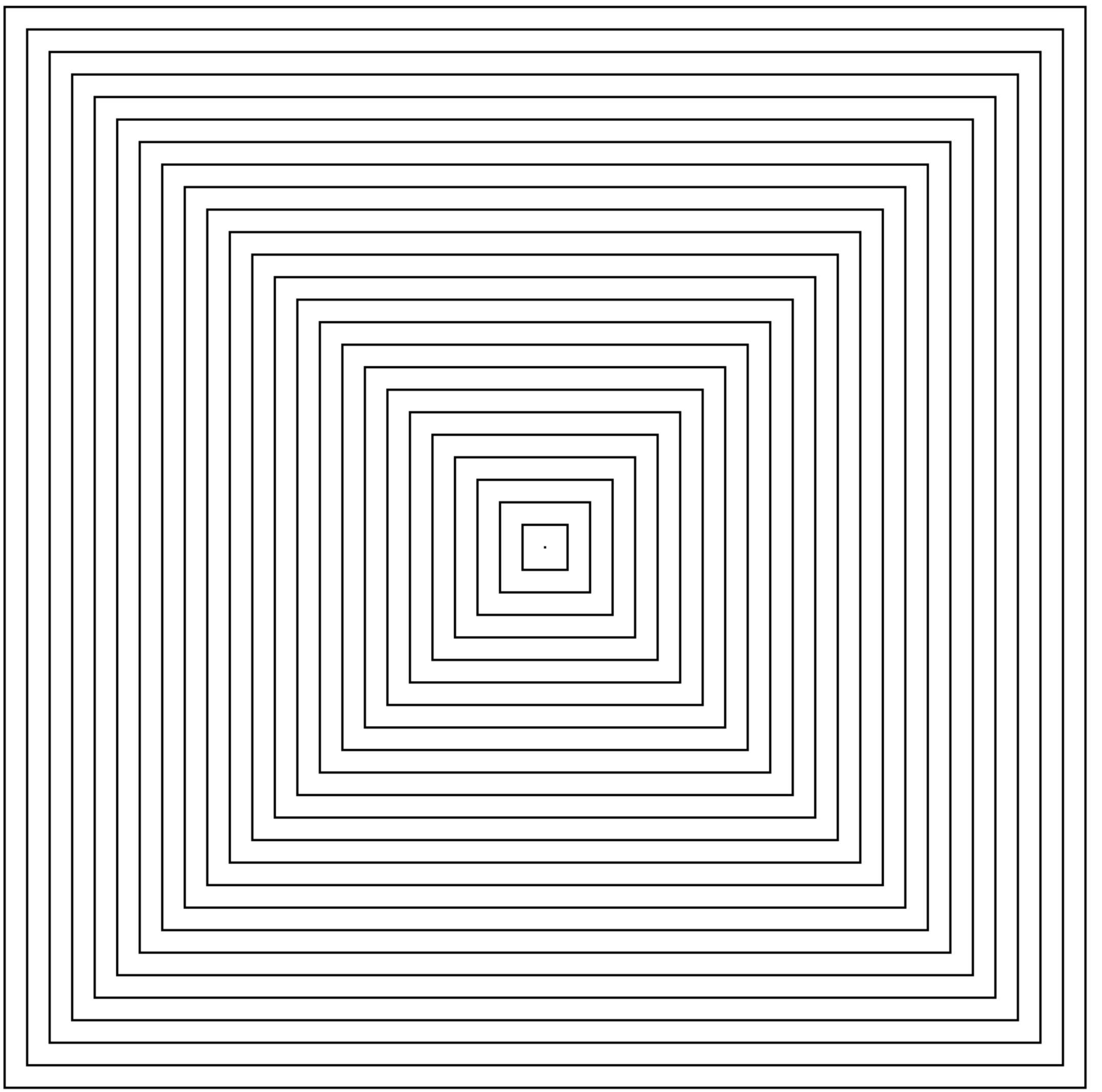
WHAT  
THE  
HECK



01:46

... vimeo

```
1 void setup() {
2   size(500,500);
3   smooth();
4   rectMode(CENTER);
5 }
6
7 void draw() {
8   background(255);
9   drawRect(width/2,height/2,width);
10  noLoop();
11 }
12
13 // Very simple function that draws one circle
14 // and recursively calls itself
15 void drawRect(float x, float y, float r) {
16   rect(x, y, r, r);
17   // Exit condition, stop when size is too small
18   if(r > 2) {
19     r -= 20;
20     // Call the function inside the function! (recursion!)
21     drawRect(x, y, r);
22   }
23 }
```



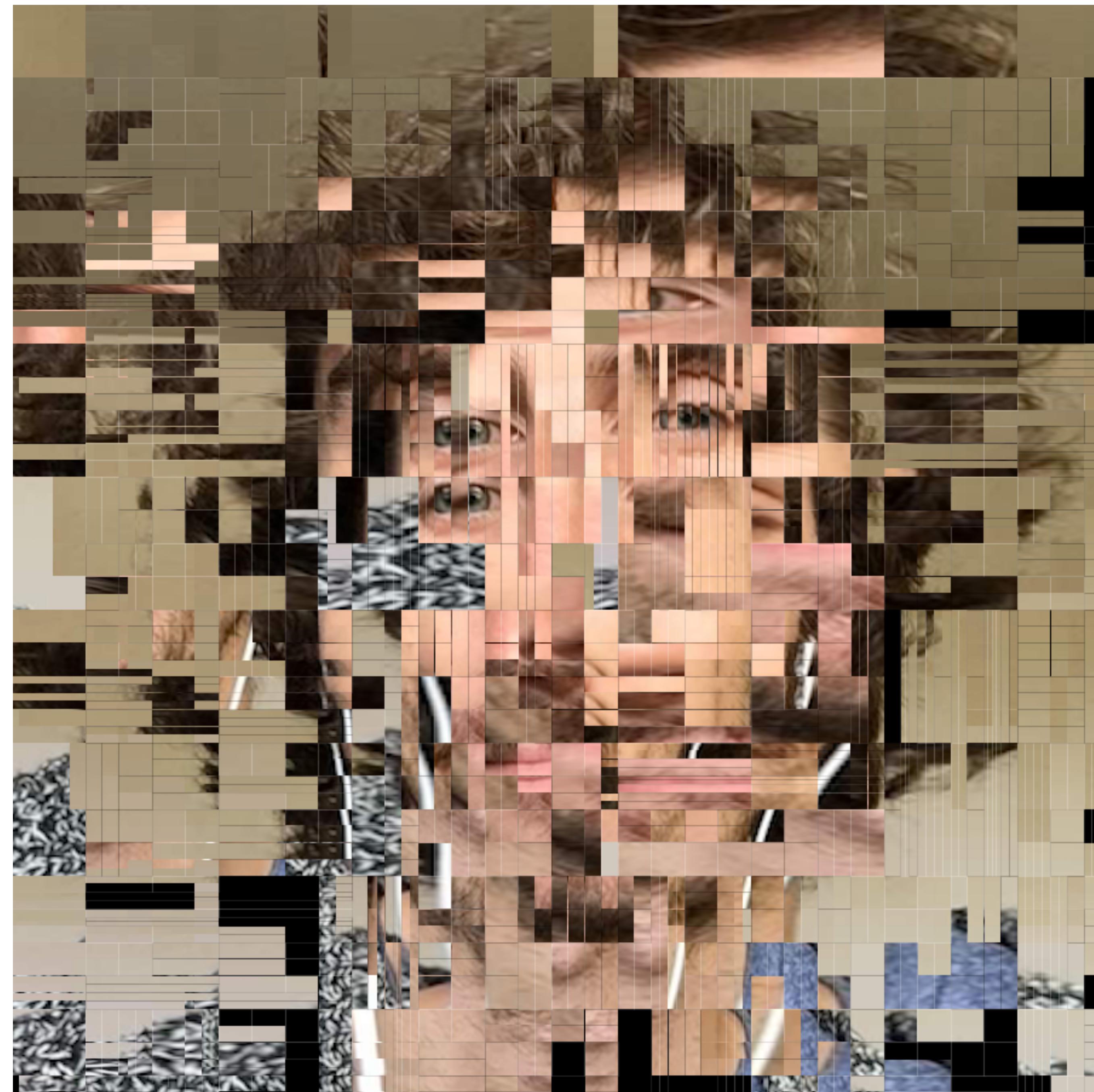
Subdivisiongrid\_recursion | Processing 4.0b8

Subdivisiongrid\_recursion

```
25 void RG(float xpos, float ypos, float gwidth, float gheight, float depth) {  
26     if (depth> 3) {  
27         //if random above threshold  
28         //recursive grid L,R  
29         if (random(1)>0.5) {  
30             fill(255);  
31             stroke(0);  
32             // divide height by 2, reduce depth by 1  
33             RG(xpos, ypos, gwidth, gheight/2, depth-1);  
34             // add height/2 to ypos, divide height by 2, reduce depth by  
35             RG(xpos, ypos+gheight/2, gwidth, gheight/2, depth-1);  
36         }  
37         //else, recursive grid U,D  
38         else {  
39             fill(0);  
40             stroke(255);  
41             // divide width by 2, reduce depth by 1  
42             // add width/2 to xpos, divide width by 2, reduce depth by  
43             RG(xpos, ypos, gwidth/2, gheight, depth-1);  
44             RG(xpos+gwidth/2, ypos, gwidth/2, gheight, depth-1);  
45         }  
46     }  
47     else {  
48         rect(xpos, ypos, gwidth, gheight);  
49     }  
50 }  
51  
52 }  
53  
54 }  
55  
56 }  
57  
58 }  
59  
60  
61  
62  
63  
64  
65 }
```

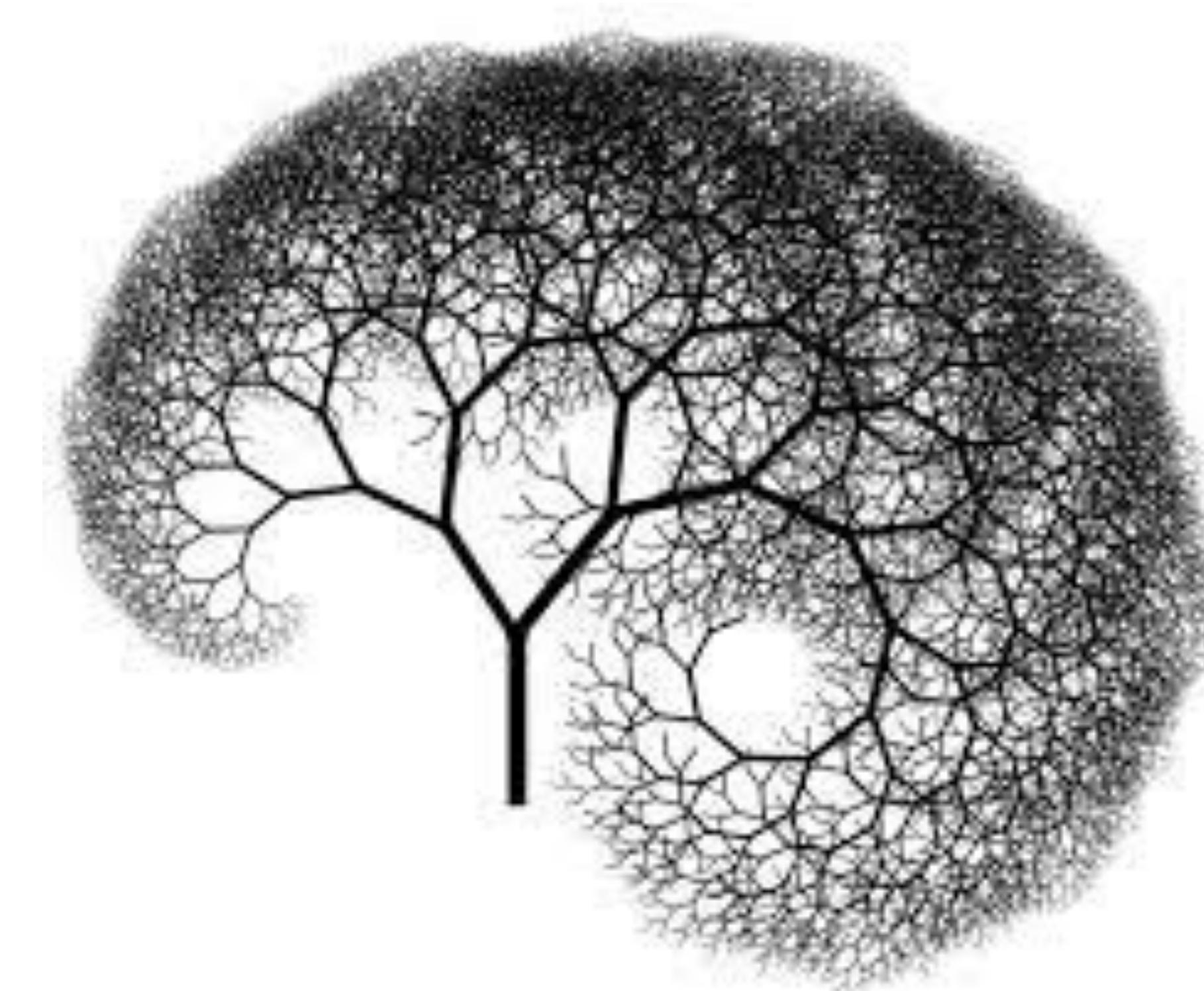
Java ▾

Subdivisiongrid\_recursion



# Fractal

A fractal is an object or quantity that displays self-similarity, in a somewhat technical sense, on all scales. The object need not exhibit exactly the same structure at all scales, but the same "type" of structures must appear on all scales.



# Eufloria



## factorial\_example



```
1 void setup() {
2     int f1 = 6;
3     int factorial1 = factorial(f1);
4
5     println(factorial1);
6 }
7
8 int factorial(int n) {
9     if (n <= 0) {
10         return 1;
11     } else {
12         println(n);
13         return n*factorial(n-1);
14     }
15 }
```

Done saving.

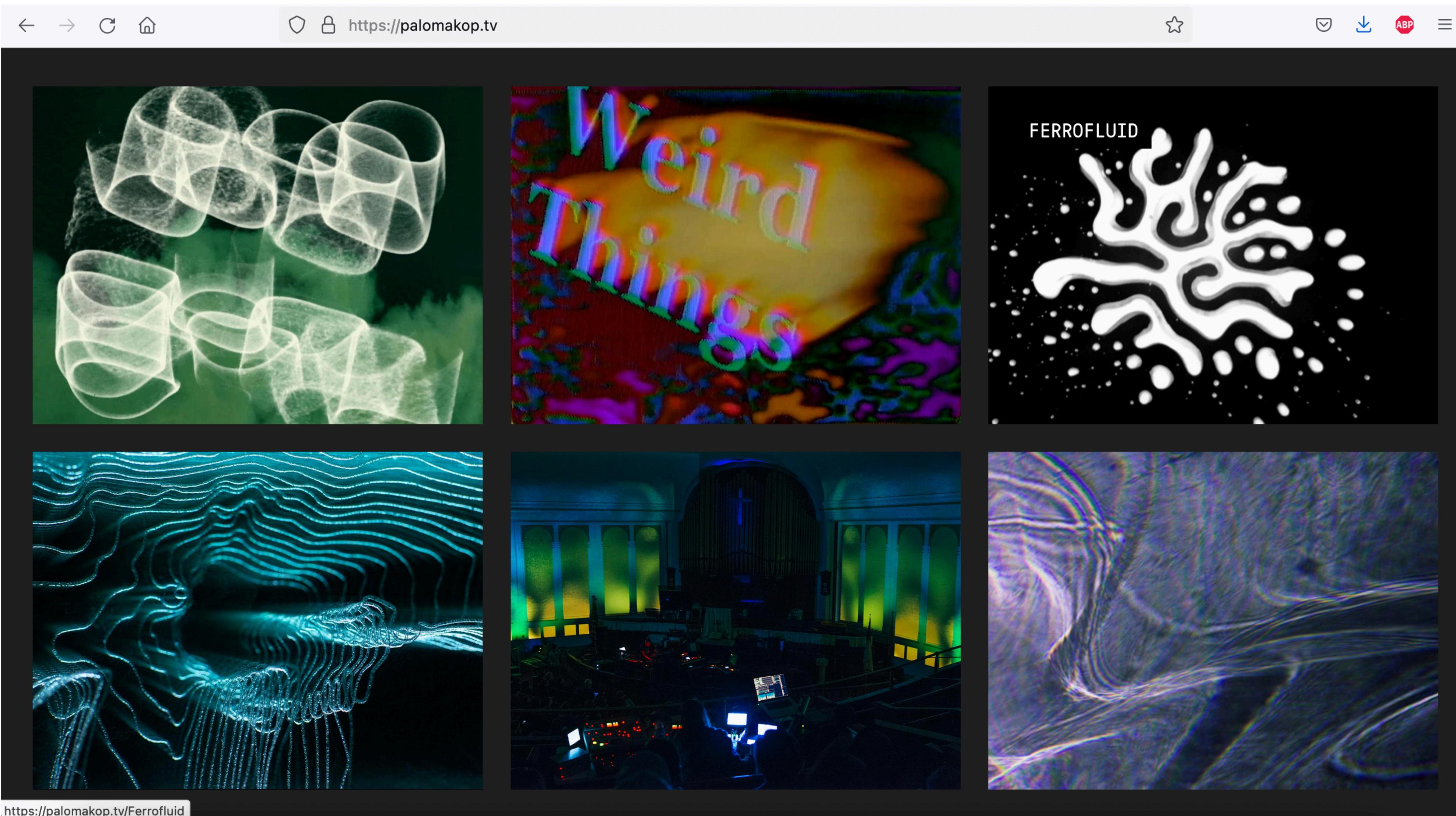
The **factorial function** (symbol: !) says to **multiply all whole numbers** from our chosen number down to 1.

Examples:

- $4! = 4 \times 3 \times 2 \times 1 = 24$
- $7! = 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 5040$
- $1! = 1$

# Paloma Kop

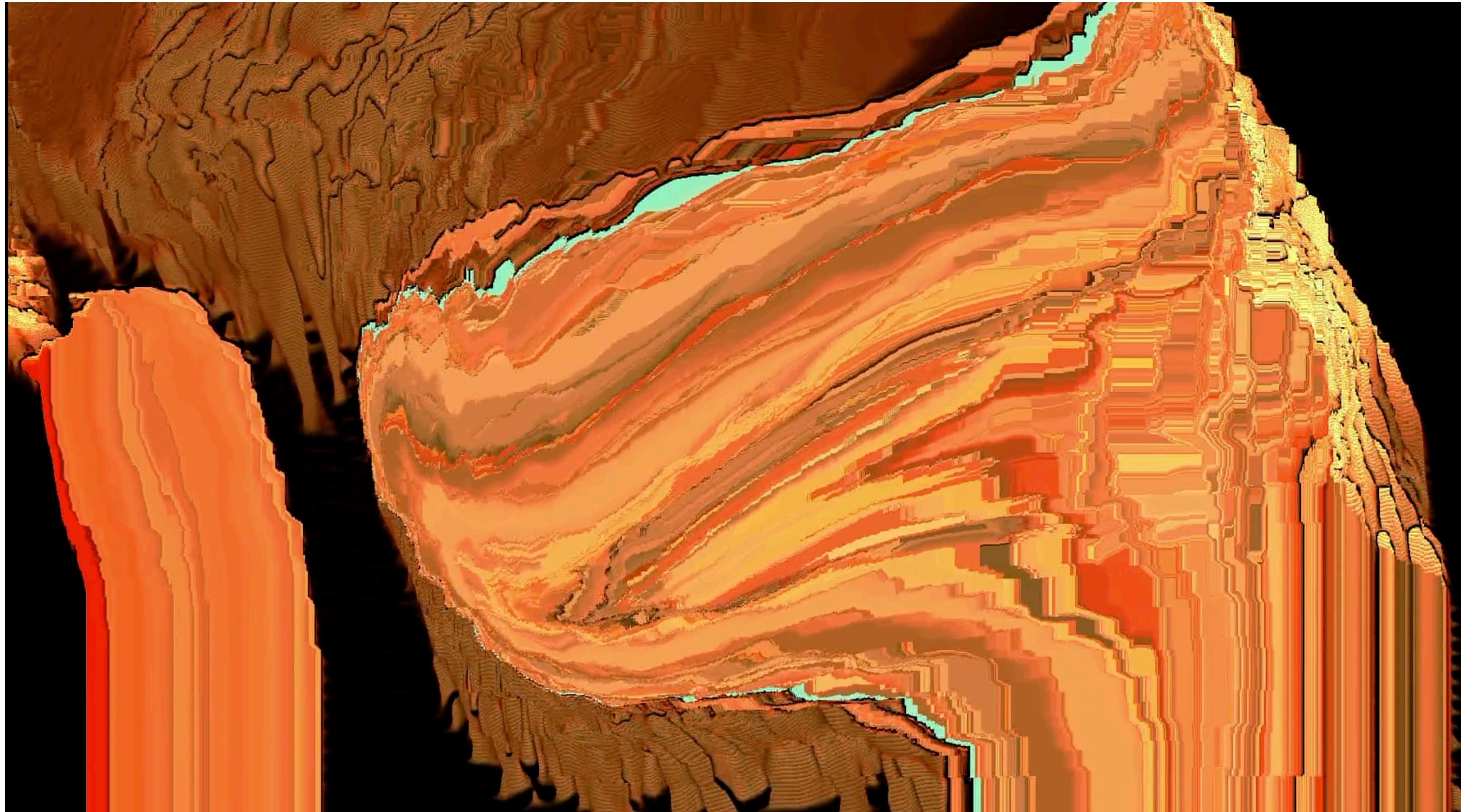
<https://palomakop.tv/Edge-of-Light>



<https://palomakop.tv/Ferrofluid>

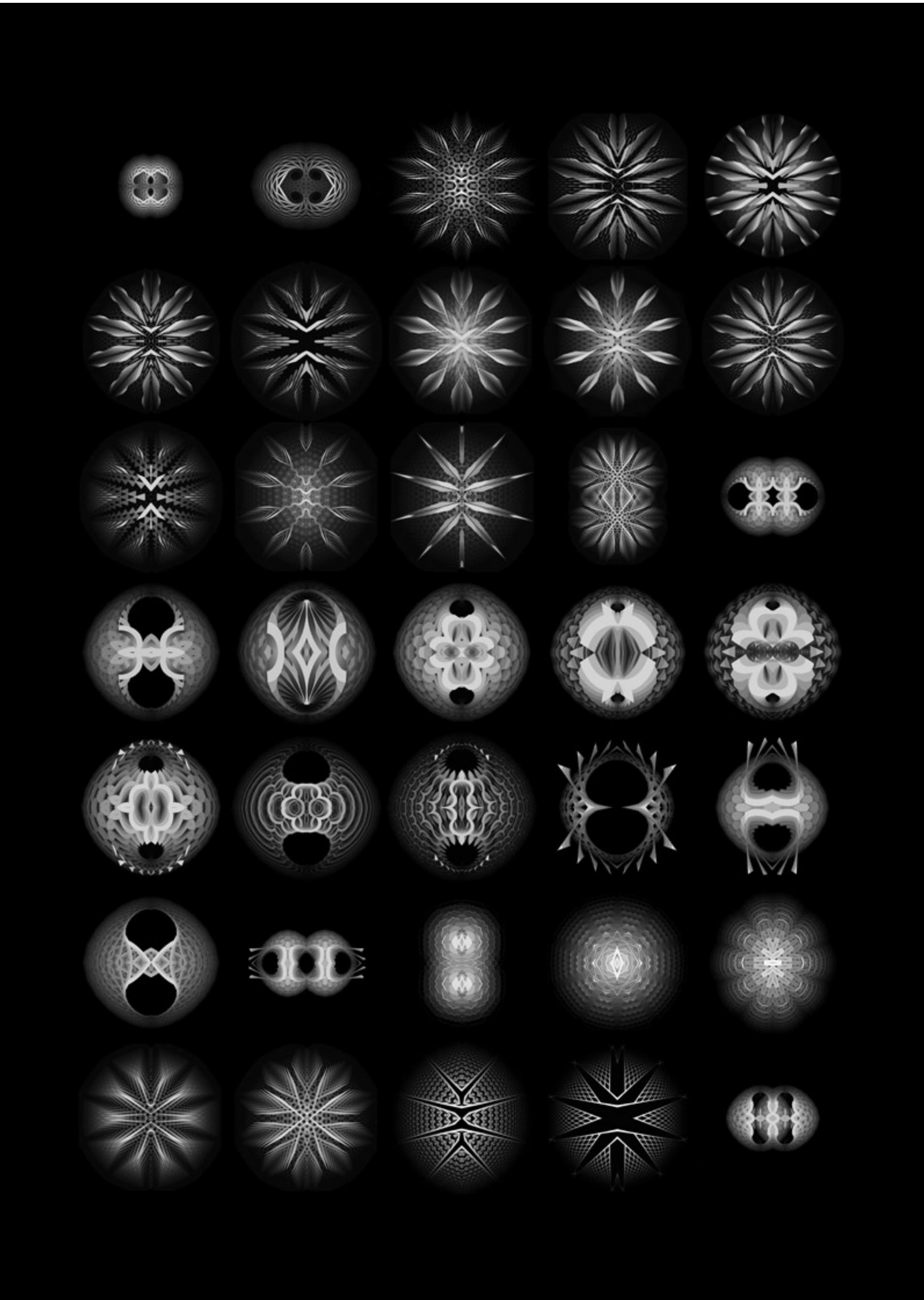
# Andrew Benson

<https://vimeo.com/179940829>



# Paul Prudence

<https://www.transphormetic.com/>  
Autotrophs-M-Sequence



# Homework

- View Andrei Jay's essay on feedback - <https://andreijaycreativecoding.com/how-does-video-feedback-work>
- Take a look at the work of Paloma Kop, Andrew Benson and Paul Prudence (linked in slides)
- Download TouchDesigner - <https://derivative.ca/download>