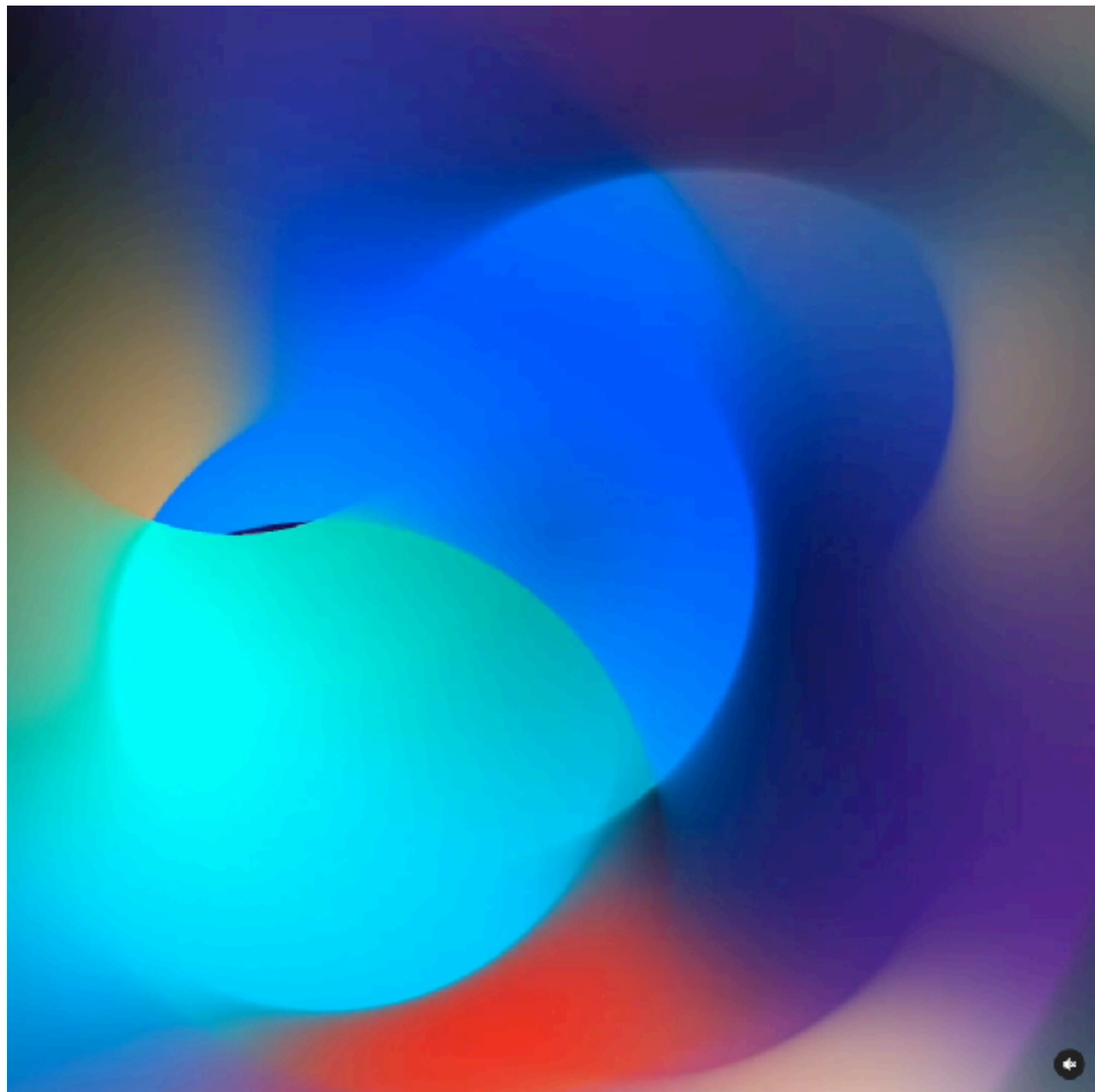


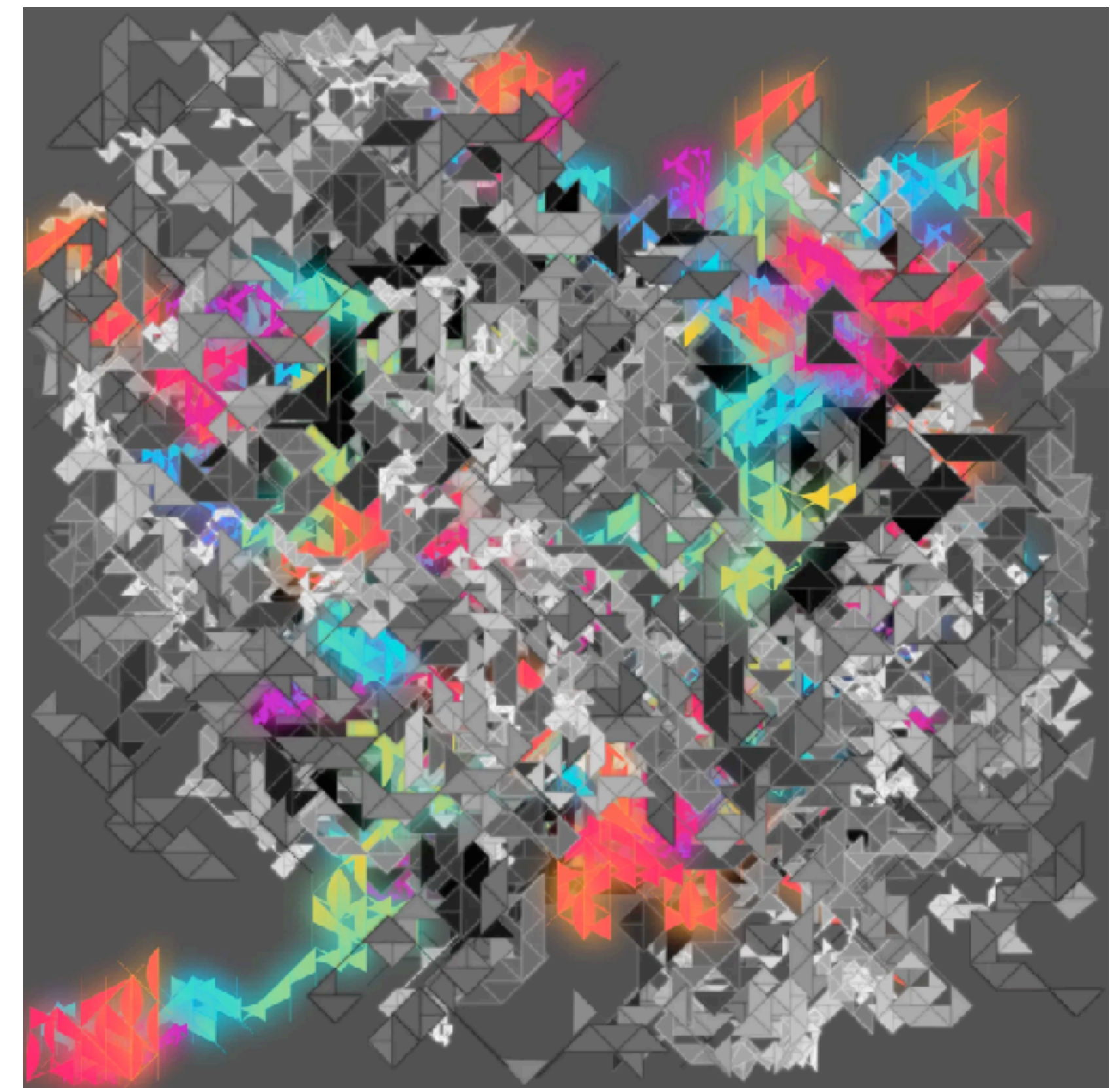
CS181DT Class 9: Creative Coding



Arcs by Zach Lieberman



Sketch Aquarium by teamLab



p5.js generative piece by shvembldr

Class 9 agenda

- Zipcrit
- PM3 artwalk
- Mini lecture: Creative coding
- Break
- Creative coding studio in p5.js

Press Fit Kit Lightning Crit (8 min)

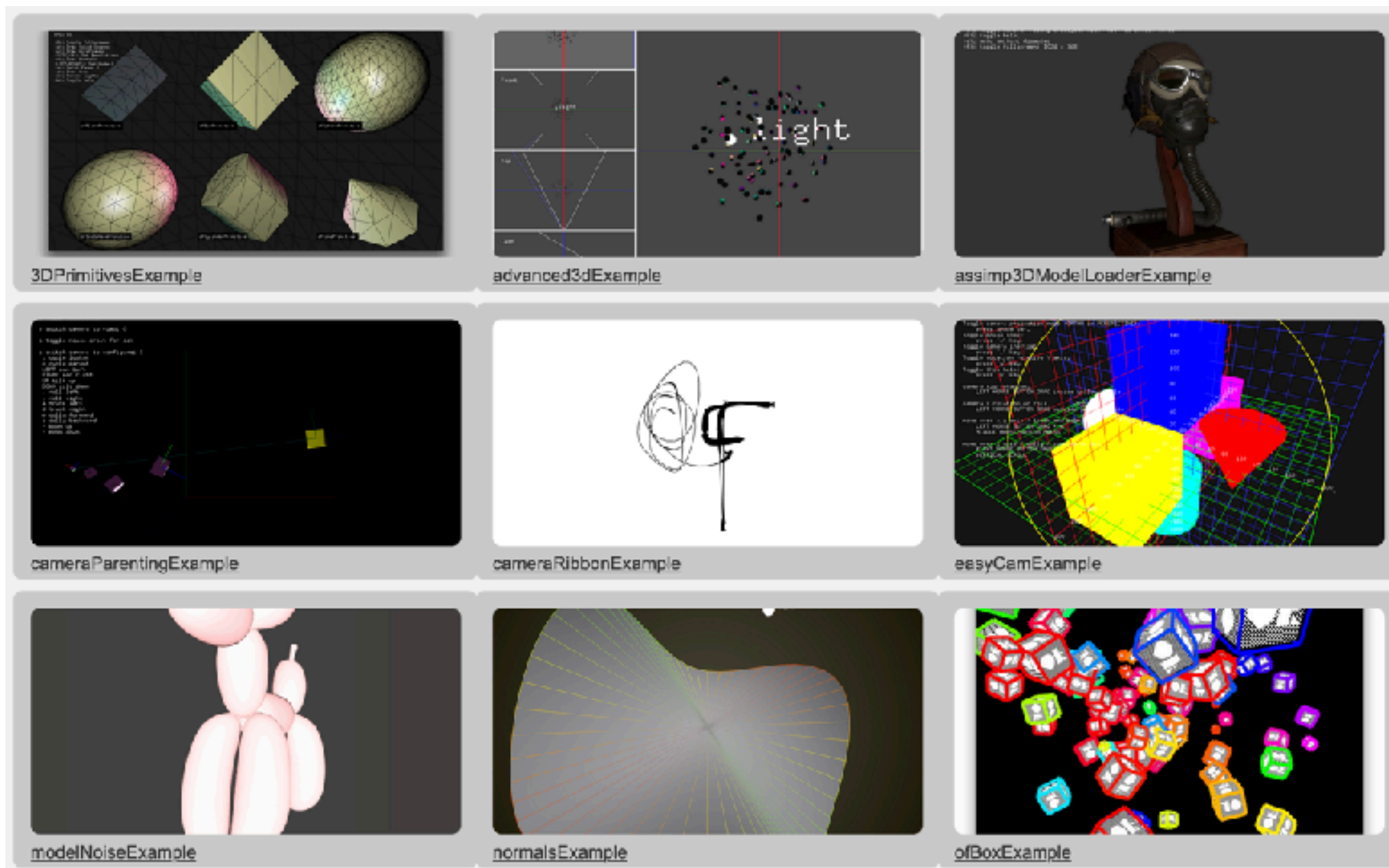
- 1 index card + 4 post-its
- On the index card, write how many sheets of new plywood you used and any messages to your audience
 - I'll collect these at the end so the course can pay the HMC makerspace :)
- 4 post-its: initial impressions for 4 pieces
 - Every piece should have at least 2 comments

Intro to Creative Coding

Creative coding

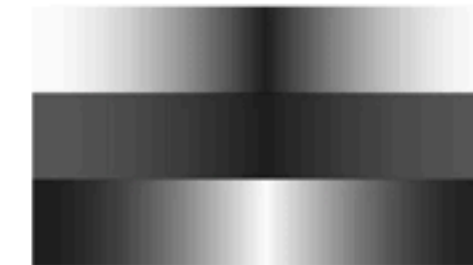
- Code written to be *expressive* rather than functional
- Many “domain specific languages” (DSLs), such as...

Compared to non-coding digital art tools (like Photoshop), what kinds of art can only be made with code?

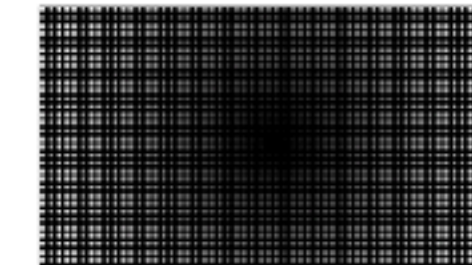


openFrameworks (C++)

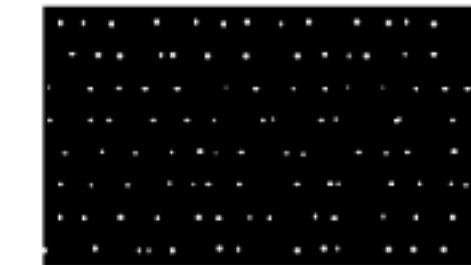
Arrays



Array

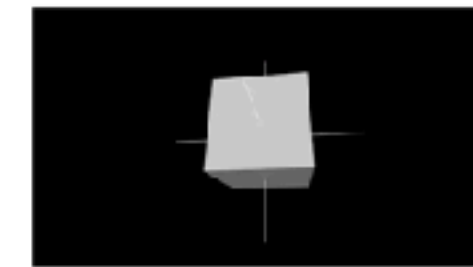


Array 2D



Array Objects

Camera



Move Eye



Orthographic

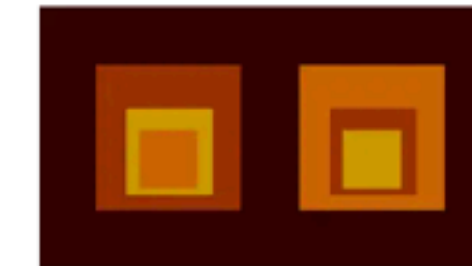


Perspective

Color



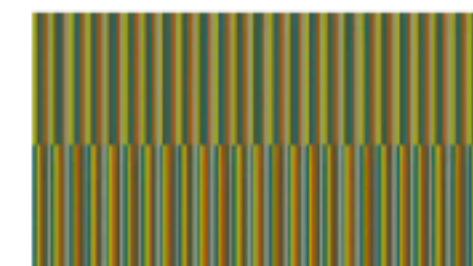
Brightness



Color Variables



Hue



Relativity



Saturation

Processing.py

p5.js

Processing
for Android

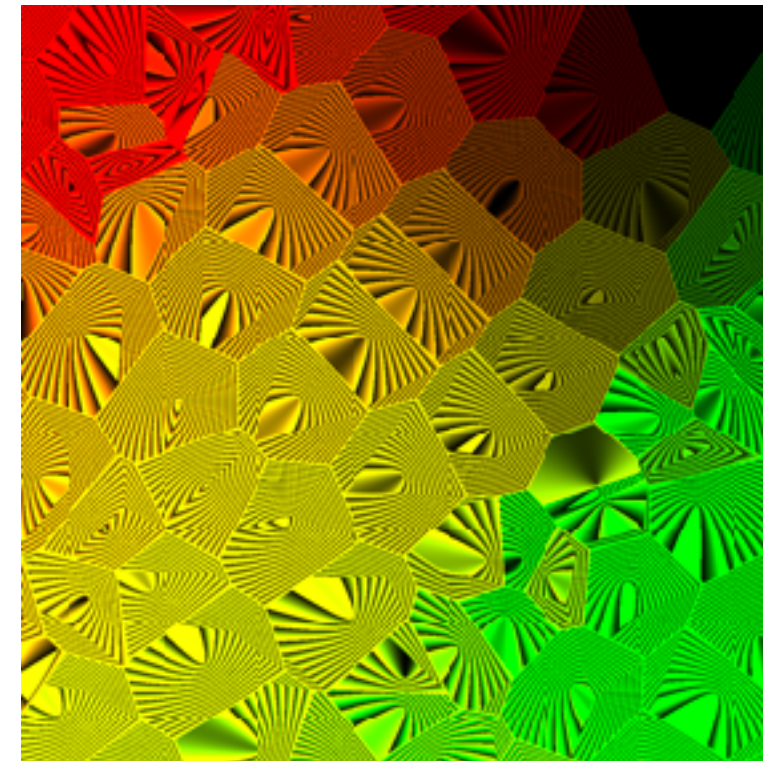
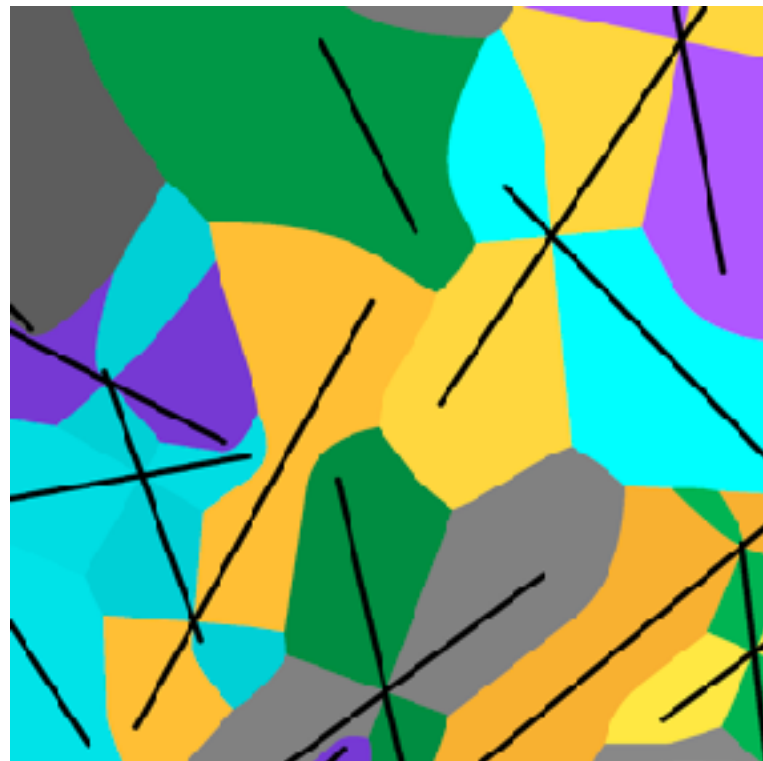
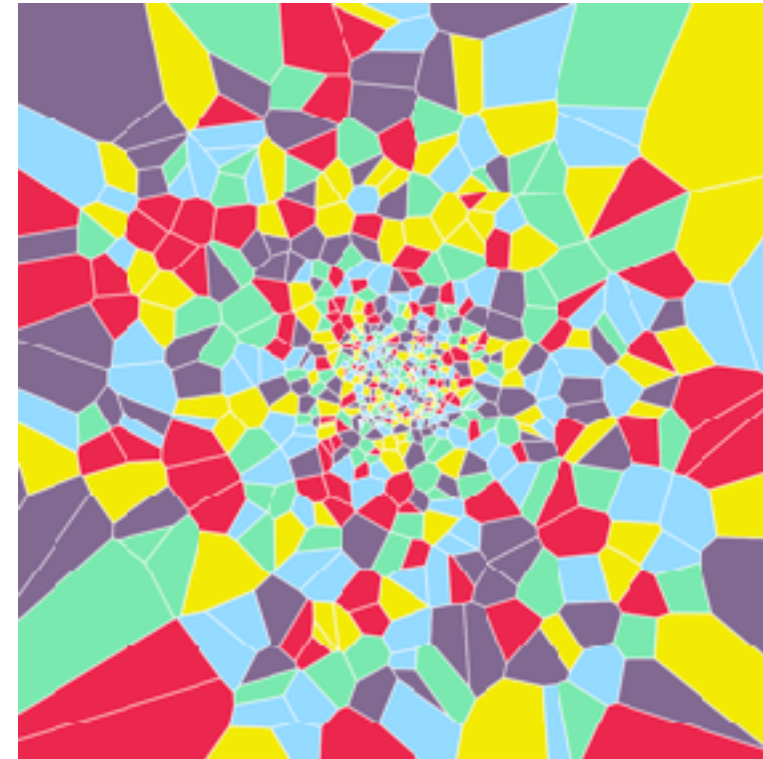
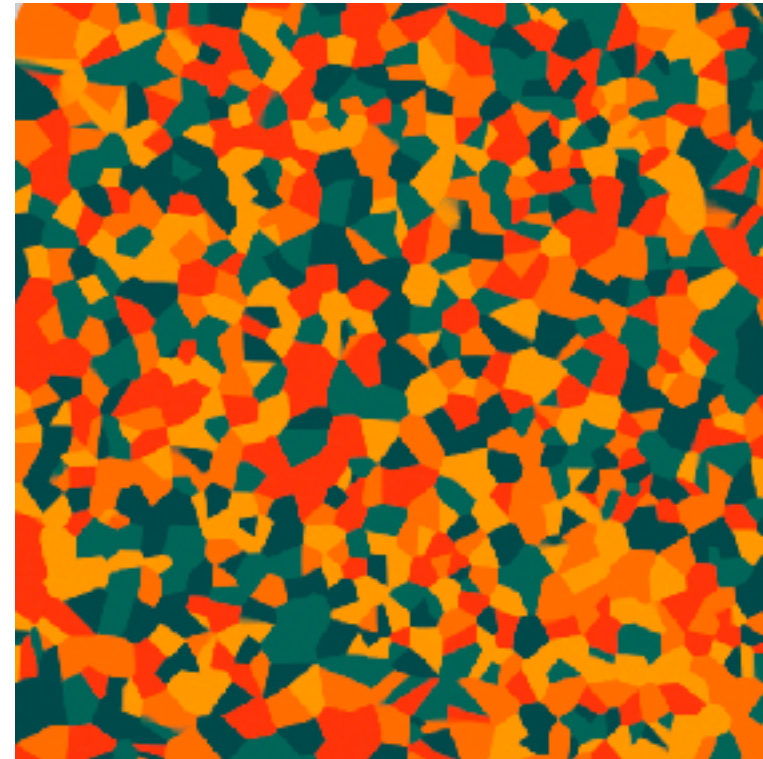
Processing
for Pi

Processing (Java)

Use cases



Interactive Art



Algorithmic art



Video Jockeying (VJing)

Community oriented

- “From the beginning, Processing was designed to be as simple as possible for beginners, knowing that its simplicity would also benefit more experienced users as well.”
- “[...] to empower people of all interests and backgrounds to learn how to program and make creative work with code, especially those who might not otherwise have access to these tools and resources.”

The screenshot shows the Processing Foundation website. At the top, there are navigation links for Processing, p5.js, Processing.py, Processing for Android, and Processing for Pi. On the right, there are social media icons for Instagram, Twitter, YouTube, Facebook, and Medium. The main content area features a purple banner for the "Processing Community Survey 2022". The text on the banner reads: "We want to hear from you! To improve our tools and community experience, we need your feedback. If you use Processing or p5.js, or any of the tools from the Processing family, this is your chance to tell us about your experience and help guide future". A deadline is noted: "Please respond by Dec 11th". Below the banner, there is a navigation bar with links for Processing, Download, Documentation, Learn, and About, along with a search bar.

Tutorials

Video Tutorials

Links to videos that cover the Processing basics.



Hello Processing
by [Daniel Shiffman et al.](#)
A quick intro guiding you to create a simple drawing tool. No software install needed.



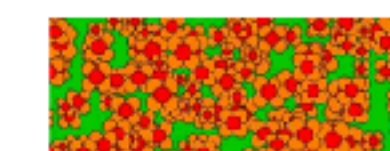
Learning Processing
by [Daniel Shiffman](#)
A complete introductory course on Processing, designed for complete beginners.



Debug
by [Daniel Shiffman](#)
Learn how to pause and step through your running code with the Processing Debugger.

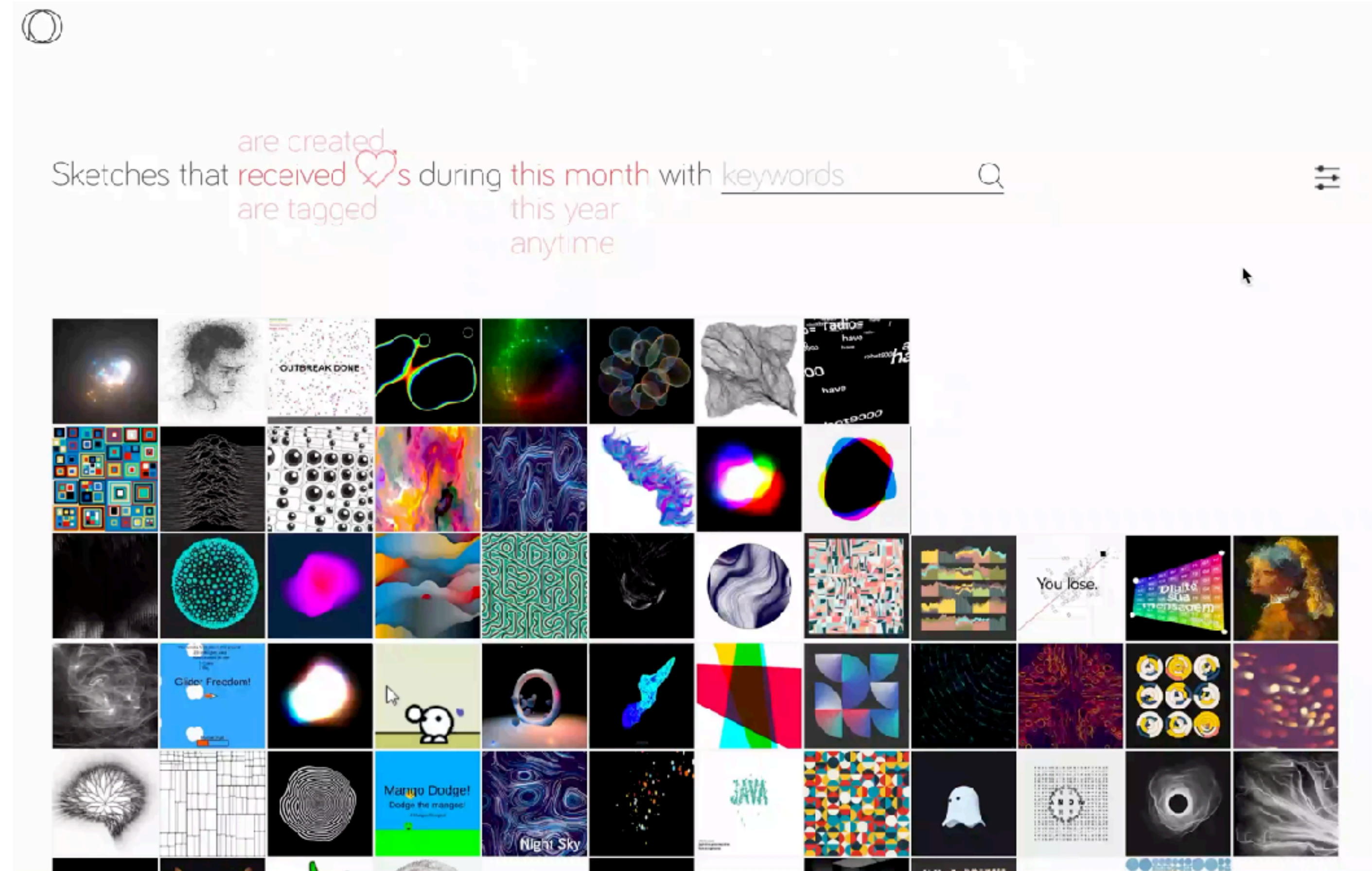


The Coding Train
by [Daniel Shiffman](#)
All aboard the Coding Train with Daniel Shiffman, a YouTube channel dedicated to beginner-friendly creative coding tutorials and challenges.



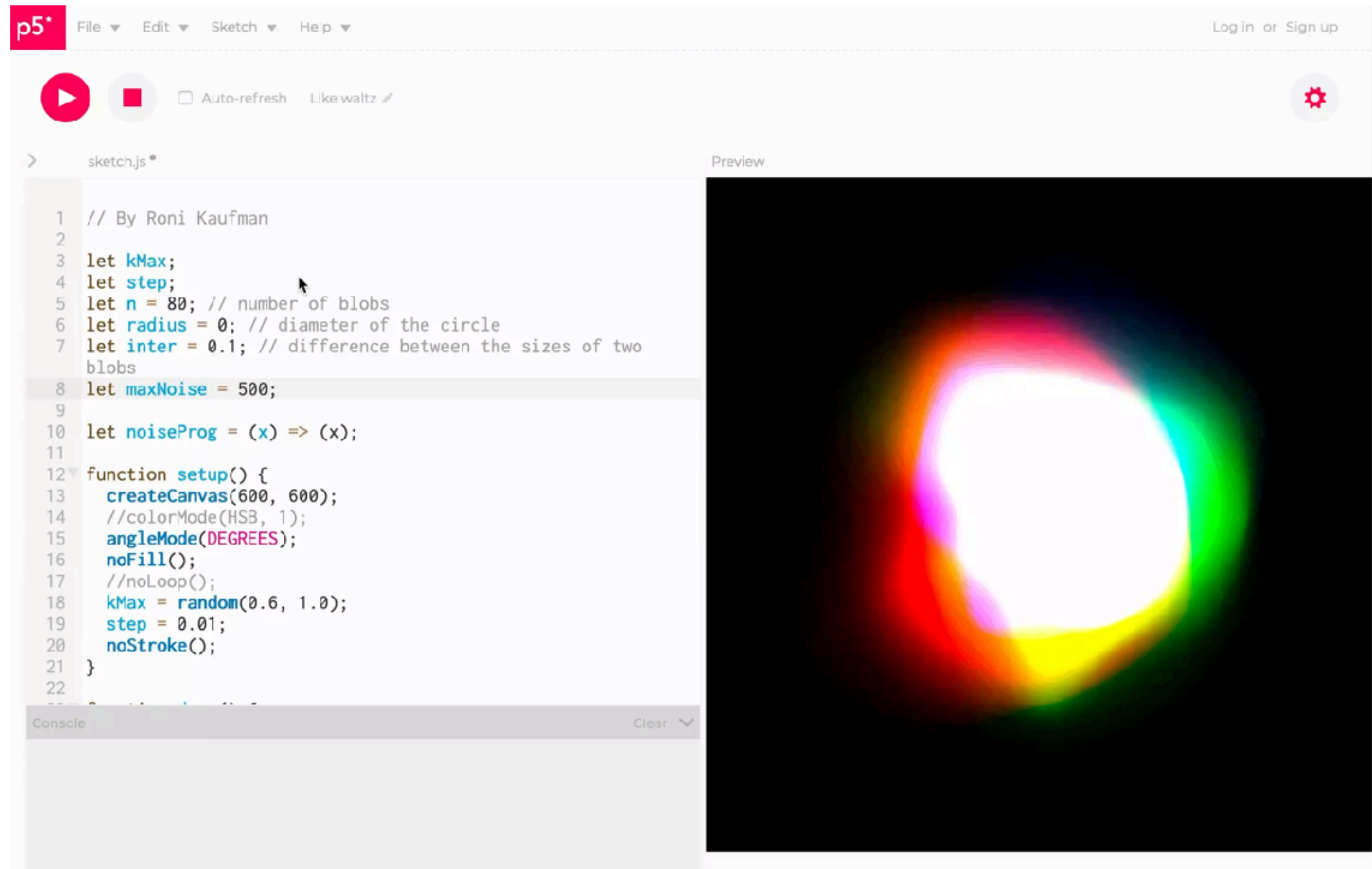
Accessible to new programmers

- Language designed in response to a frustration of the cultural elitism of programming
- Learning through editing lots of community made examples and tutorials, or in person workshops



Accessible to new programmers

- Language designed in response to a frustration of the cultural elitism of programming
- Learning through editing lots of community made examples and tutorials, or in person workshops



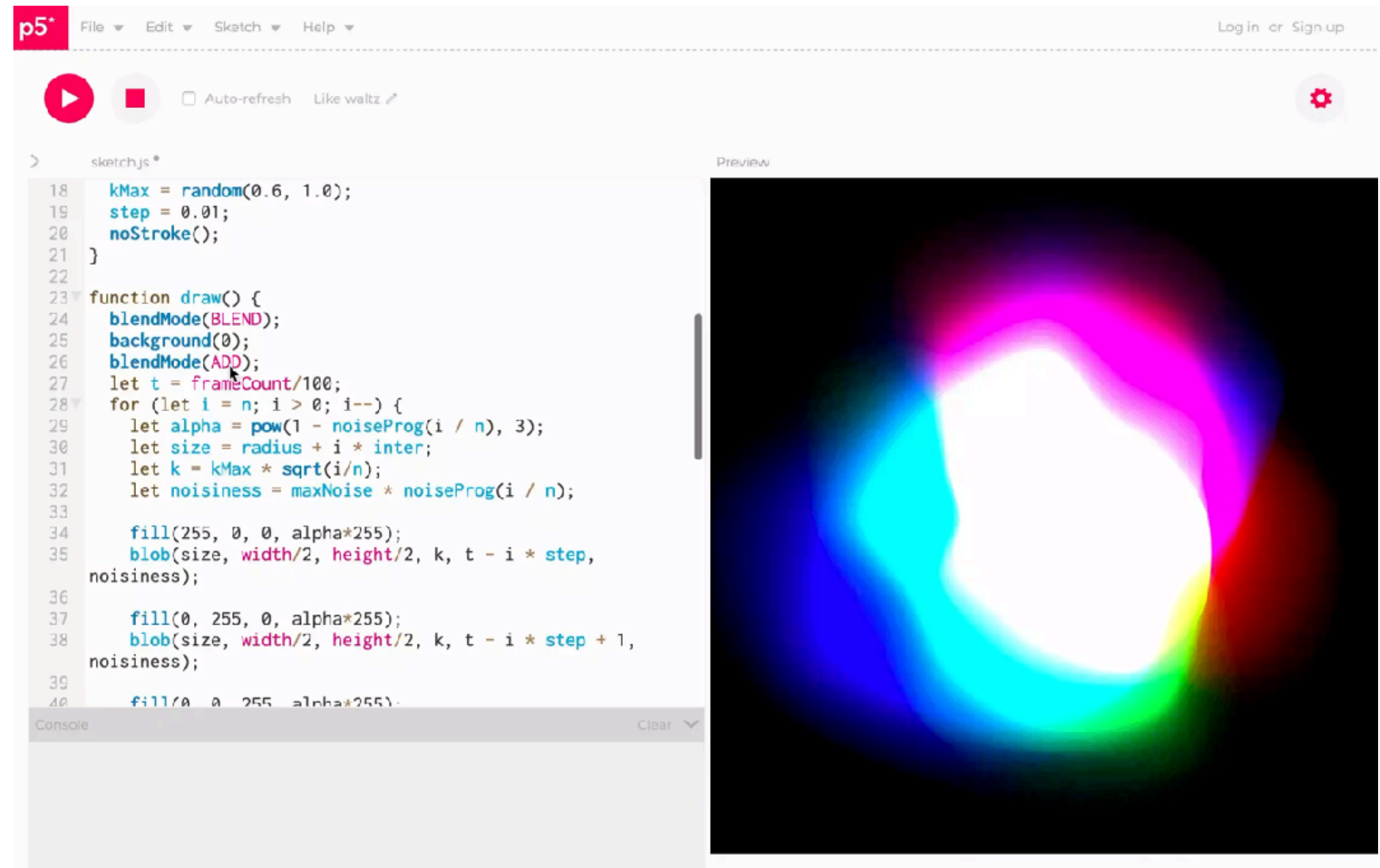
The screenshot shows the p5.js web editor interface. The top bar includes the p5.js logo, a menu (File, Edit, Sketch, Help), and a 'Log in or Sign up' link. Below the menu are a play button, a stop button, and a checkbox for 'Auto-refresh'. The main area is split into a code editor on the left and a preview window on the right. The code editor shows the following JavaScript code:

```
1 // By Roni Kaufman
2
3 let kMax;
4 let step;
5 let n = 80; // number of blobs
6 let radius = 0; // diameter of the circle
7 let inter = 0.1; // difference between the sizes of two
  blobs
8 let maxNoise = 500;
9
10 let noiseProg = (x) => (x);
11
12 function setup() {
13   createCanvas(600, 600);
14   //colorMode(HSB, 1);
15   angleMode(DEGREES);
16   noFill();
17   //noLoop();
18   kMax = random(0.6, 1.0);
19   step = 0.01;
20   noStroke();
21 }
22
```

The preview window on the right shows a colorful, glowing ring or blob shape on a black background. The ring is composed of many small, overlapping circles in various colors, creating a vibrant, multi-colored effect.

Accessible to new programmers, but different than manually making art

- Language designed in response to a frustration of the cultural elitism of programming
- Learning through editing lots of community made examples and tutorials, or in person workshops



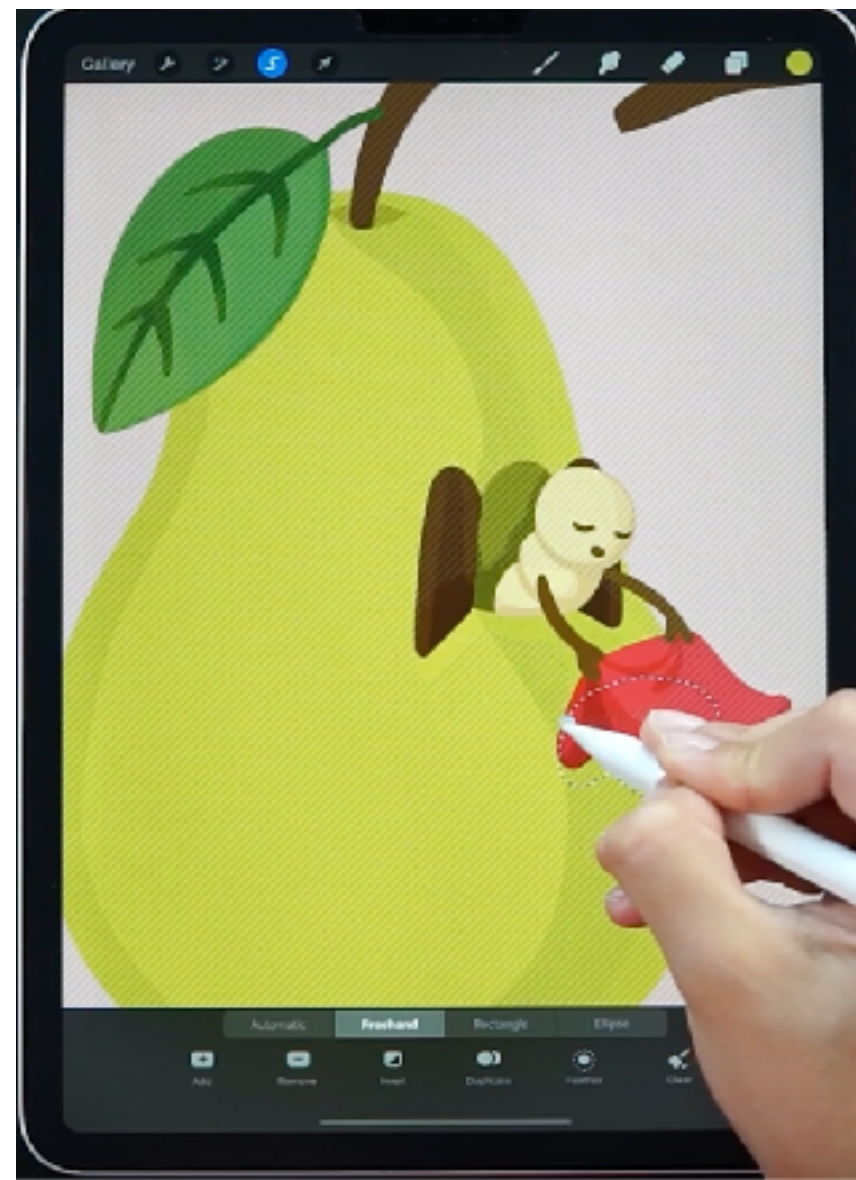
The screenshot shows the p5.js web editor interface. The top navigation bar includes 'p5*', 'File', 'Edit', 'Sketch', 'Help', and 'Login or Sign up'. Below the navigation bar are control buttons: a play button, a stop button, 'Auto-refresh', and 'Like waltz'. The main editor area is split into two panes. The left pane, titled 'sketch.js', contains the following code:

```
18 kMax = random(0.6, 1.0);
19 step = 0.01;
20 noStroke();
21 }
22
23 function draw() {
24   blendMode(BLEND);
25   background(0);
26   blendMode(ADD);
27   let t = frameCount/100;
28   for (let i = n; i > 0; i--) {
29     let alpha = pow(1 - noiseProg(i / n), 3);
30     let size = radius + i * inter;
31     let k = kMax * sqrt(i/n);
32     let noisiness = maxNoise * noiseProg(i / n);
33
34     fill(255, 0, 0, alpha*255);
35     blob(size, width/2, height/2, k, t - i * step,
36         noisiness);
37
38     fill(0, 255, 0, alpha*255);
39     blob(size, width/2, height/2, k, t - i * step + 1,
40         noisiness);
41     fill(0, 0, 255, alpha*255);
```

The right pane, titled 'Preview', shows a colorful, abstract visualization of the code. It features a central white shape surrounded by a thick, multi-colored ring. The colors transition from cyan on the left, through green and yellow, to magenta and red on the right. The background is black.



Artwork



Canvas

- Directly manipulate the output
- Immediately observe how actions result in changes
- Allows for open-ended exploration



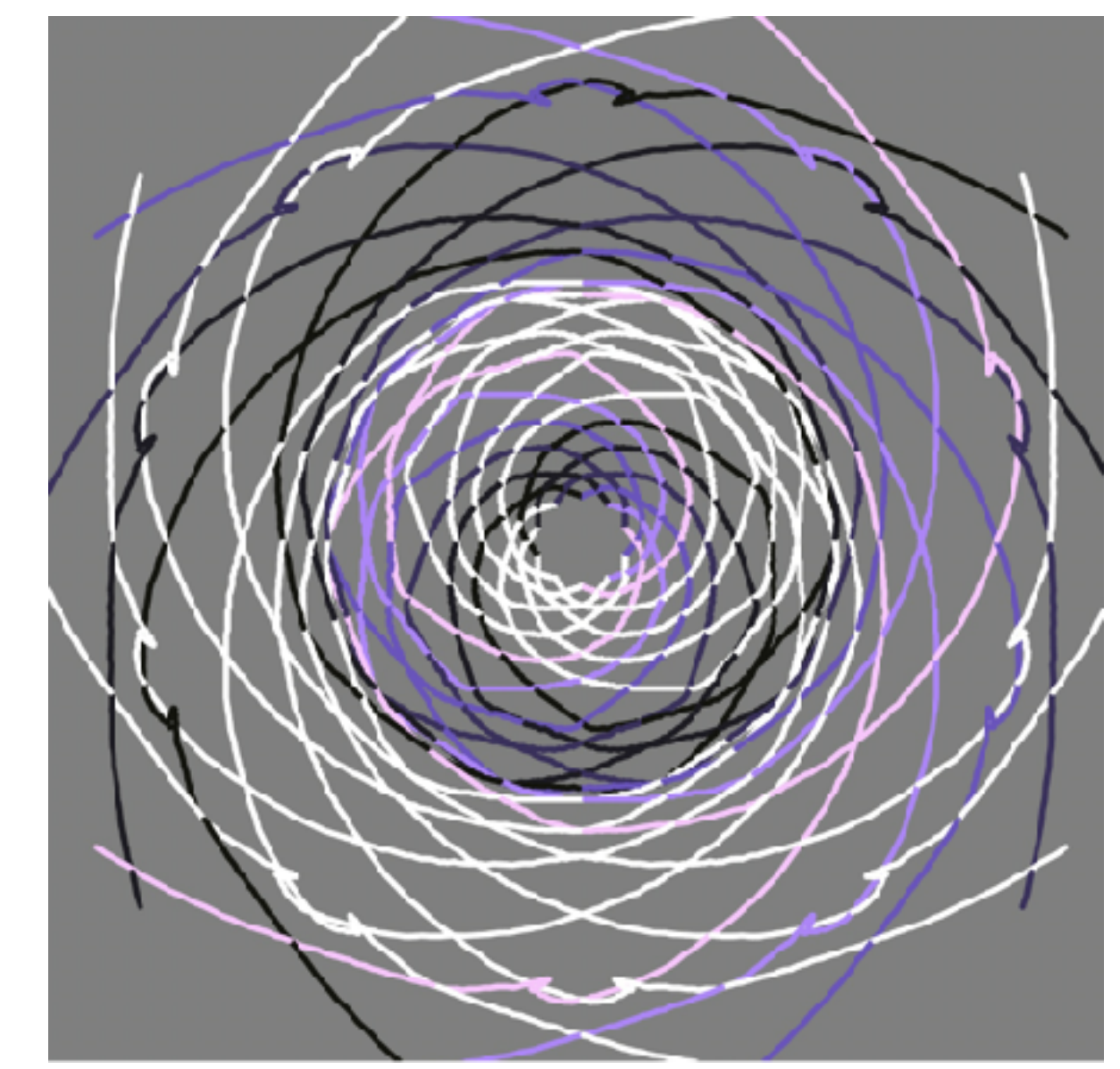
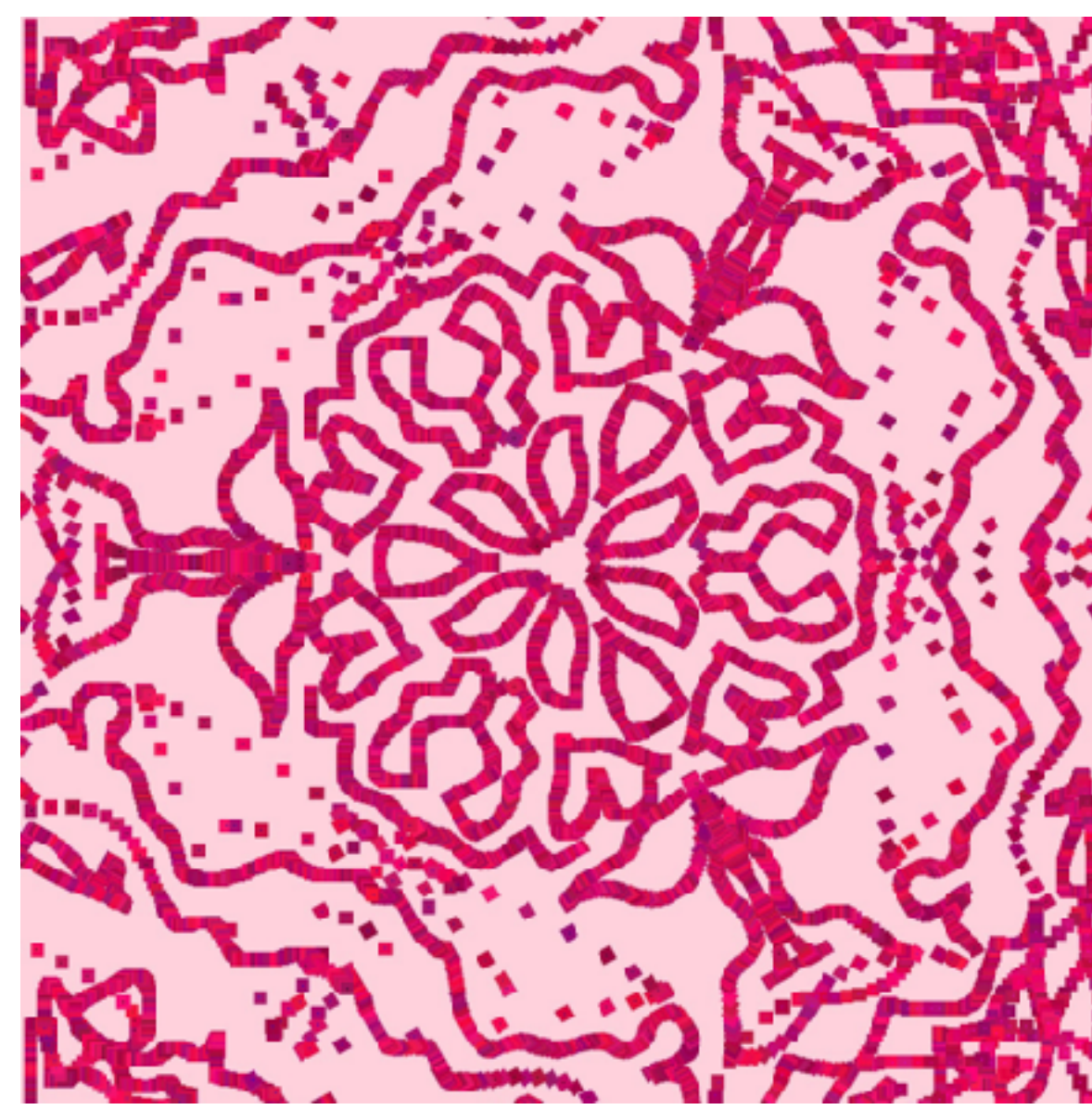
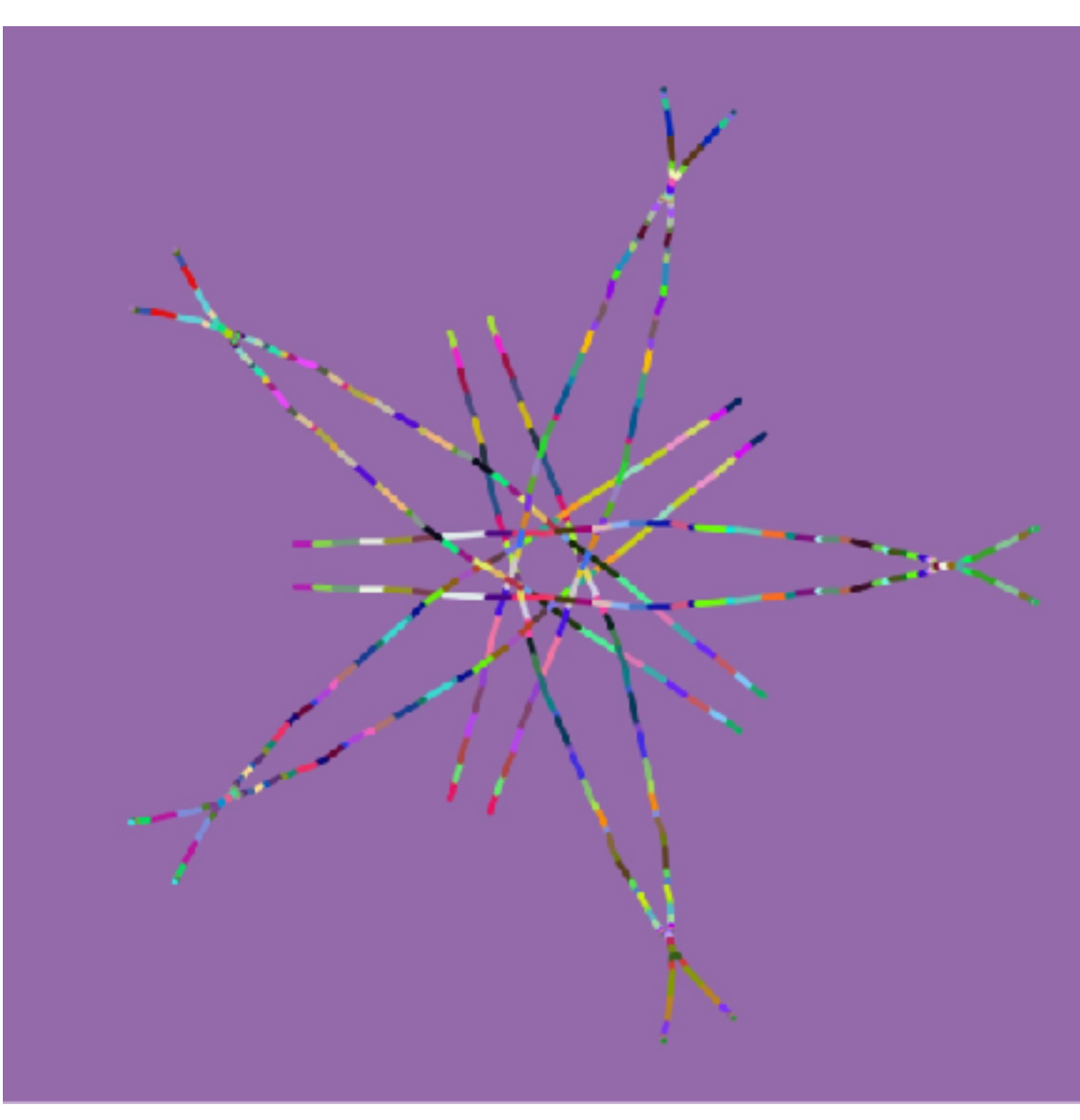
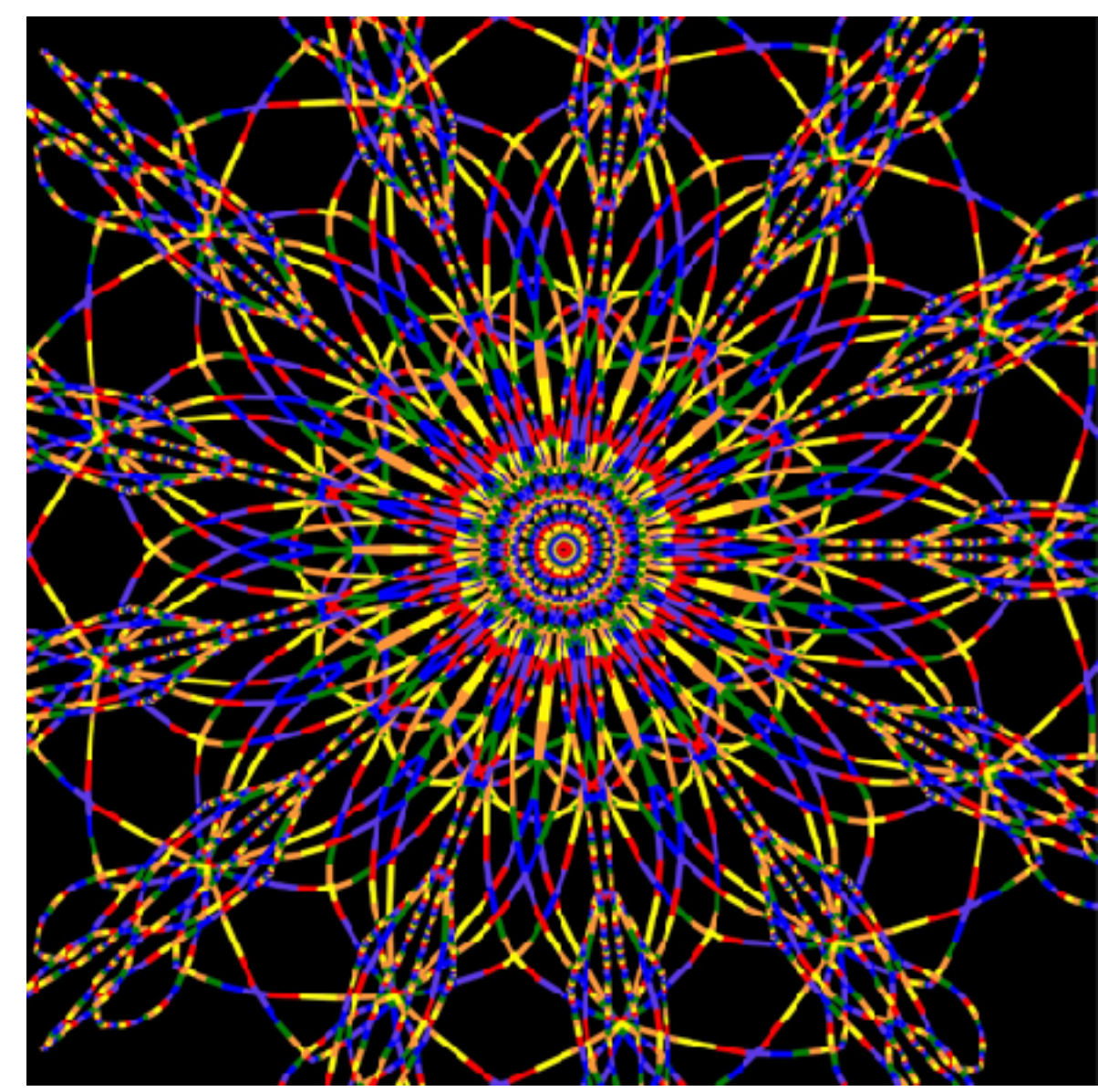
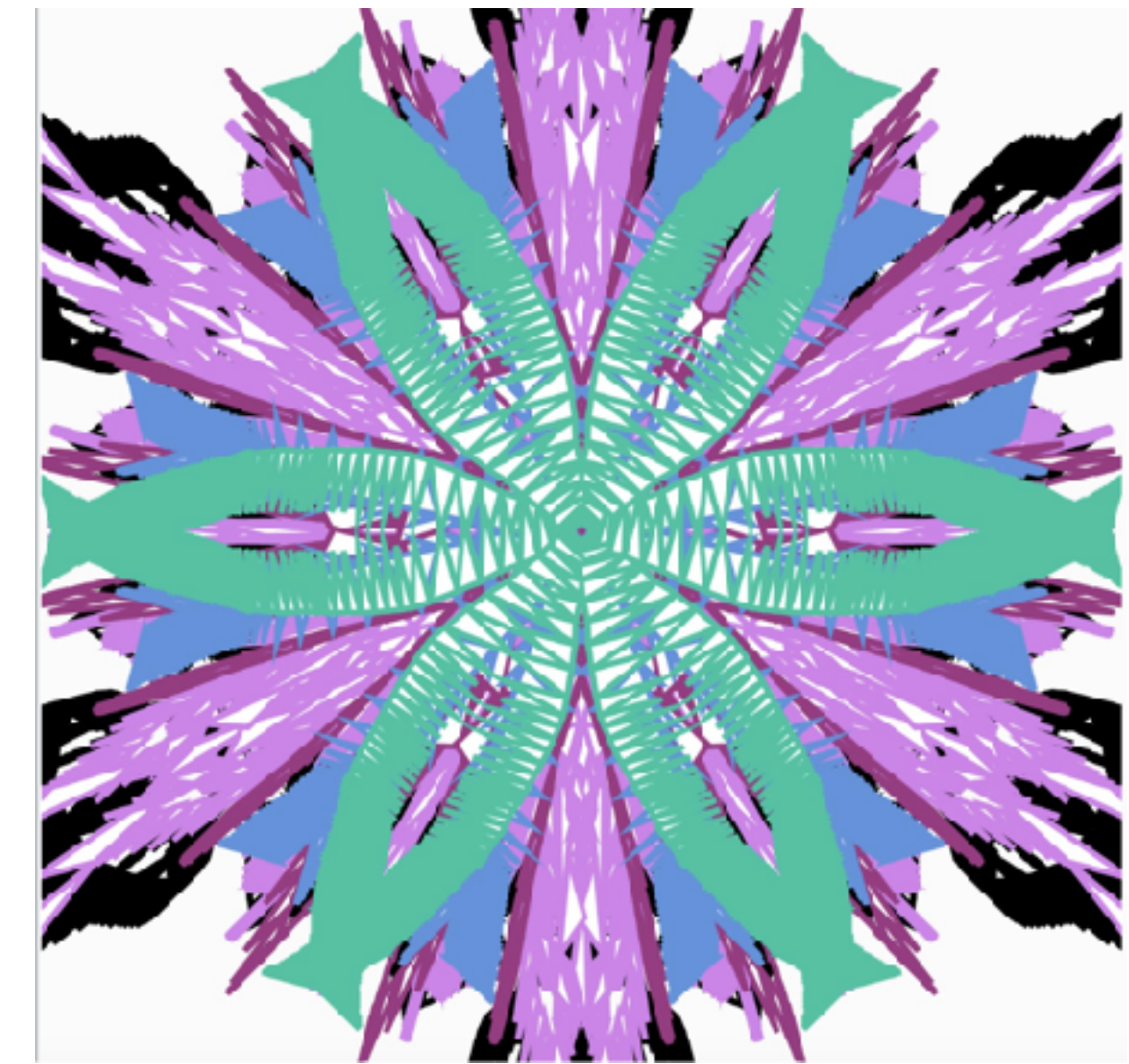
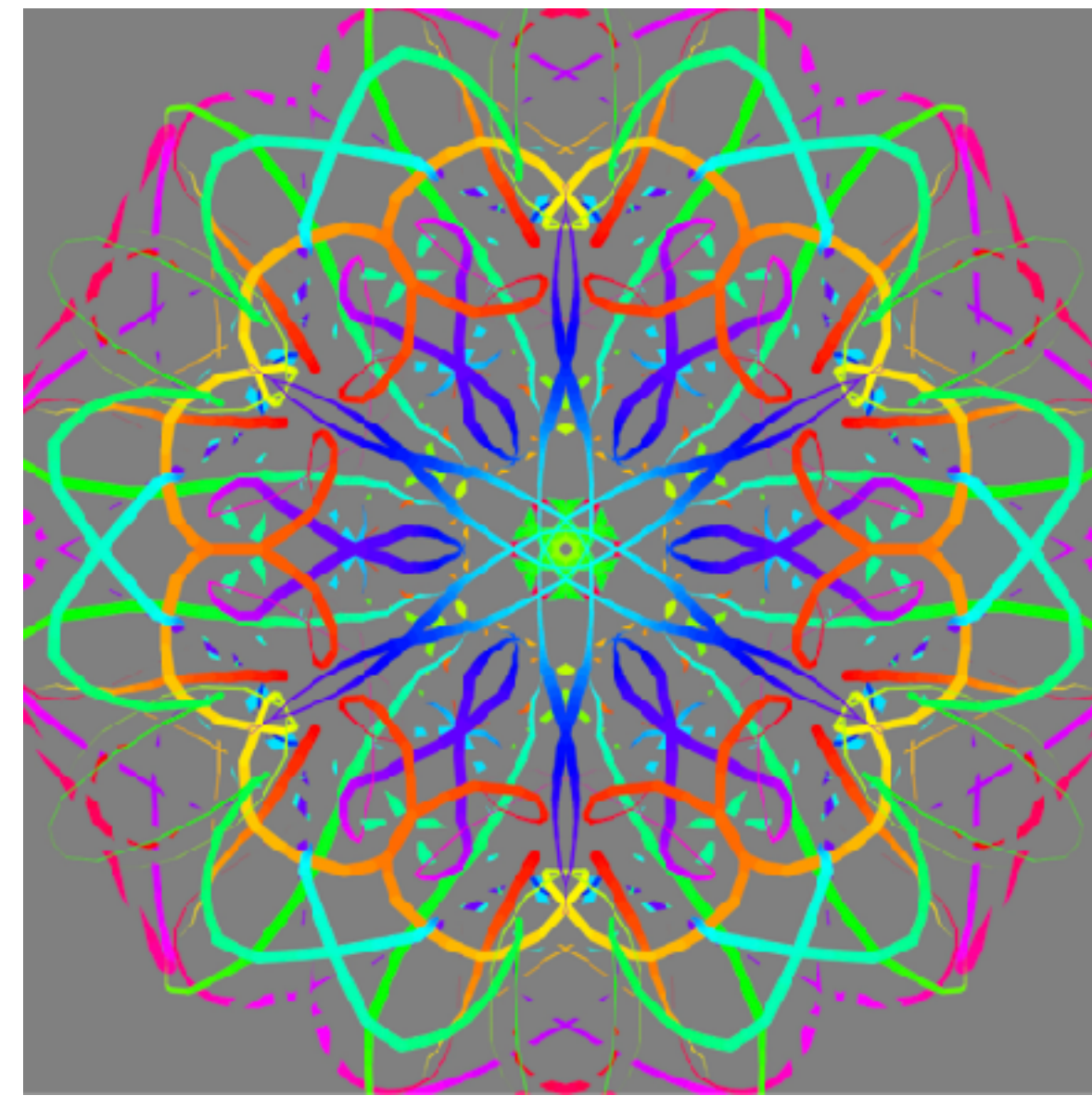
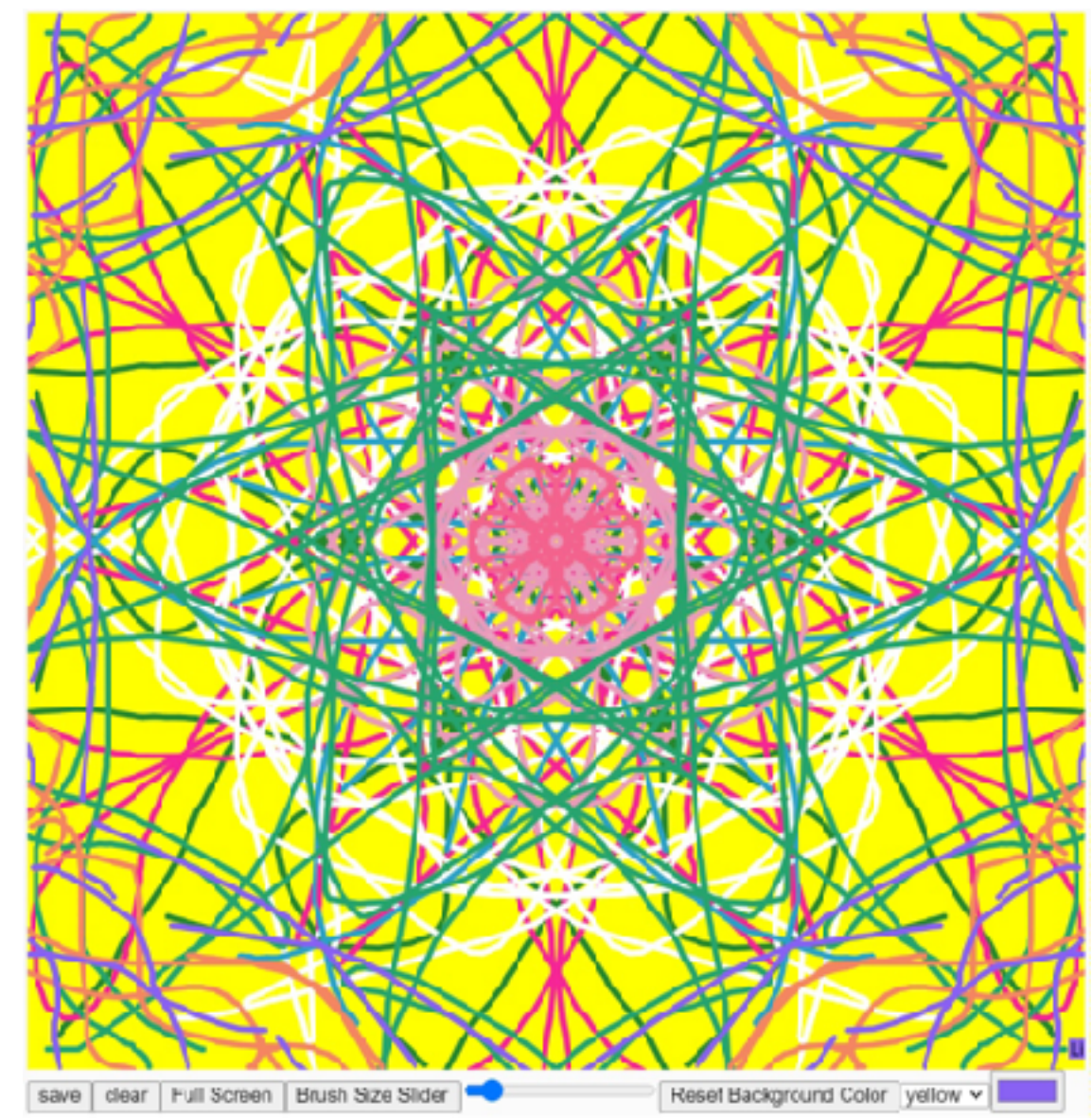
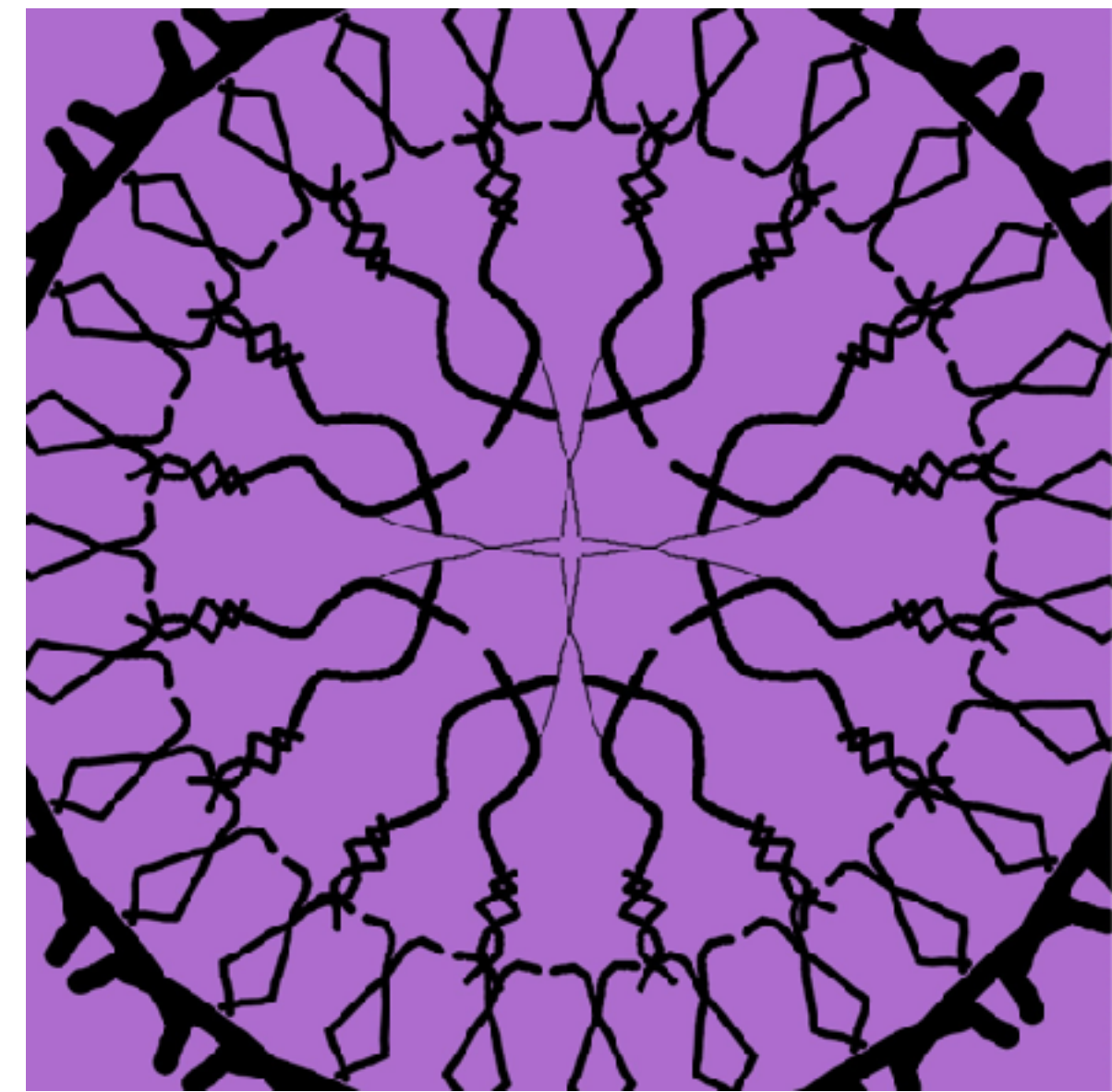
Artwork



Canvas

- Manipulate abstract symbols (code)
- Programming and execution are separate, unclear which pixel is caused by which line of code
- Requires more linear structure and building blocks before exploring

Your turn: p5.js studio



p5.js is a friendly tool for learning to code and make art. It is a free and open-source JavaScript library built by an inclusive, nurturing community. p5.js welcomes artists, designers, beginners, educators, and anyone else!



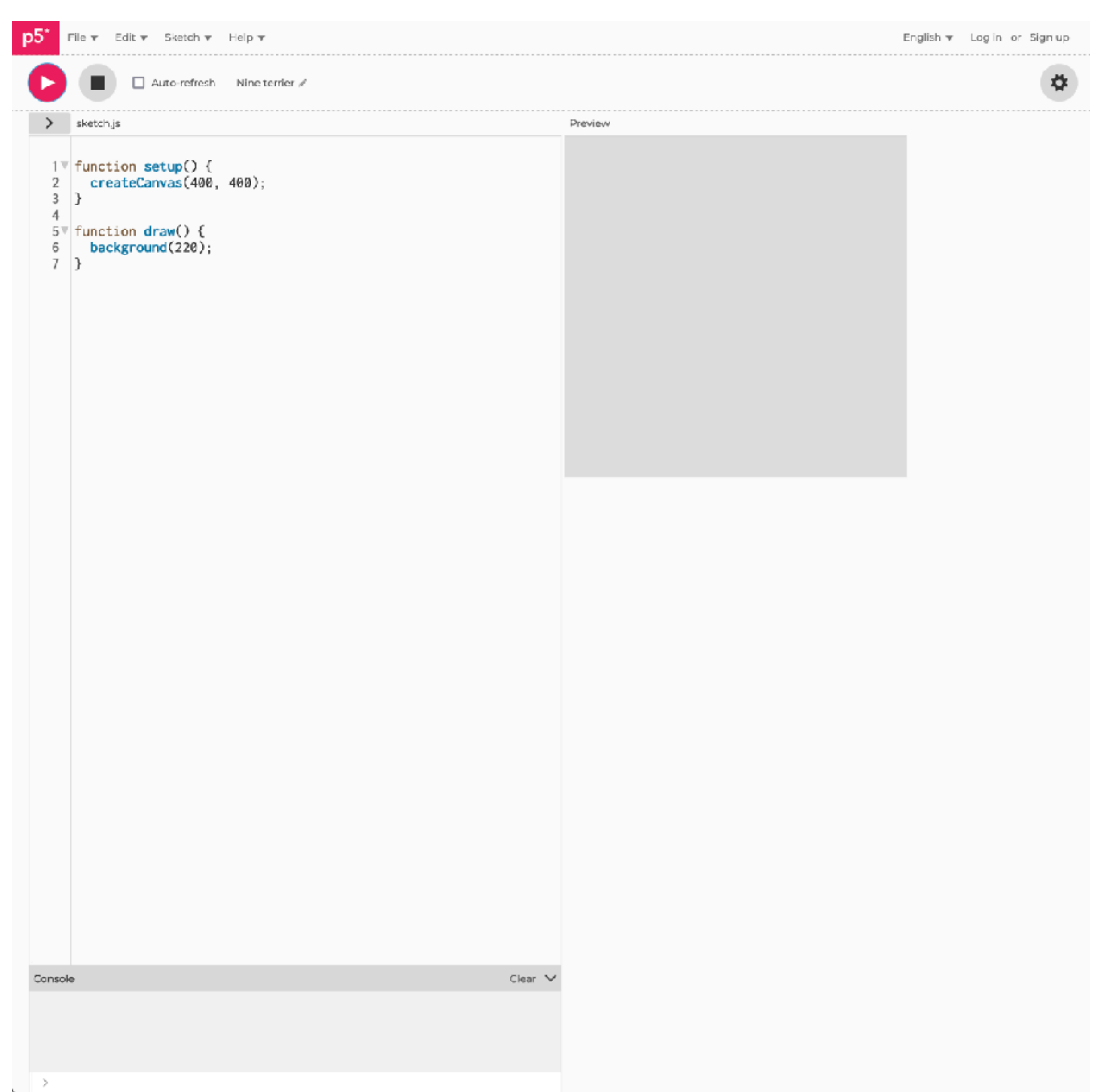
Coding Club for people aged 50+ in Korea, led by Inhwa Yeom.



</> Start Coding

♥ Donate

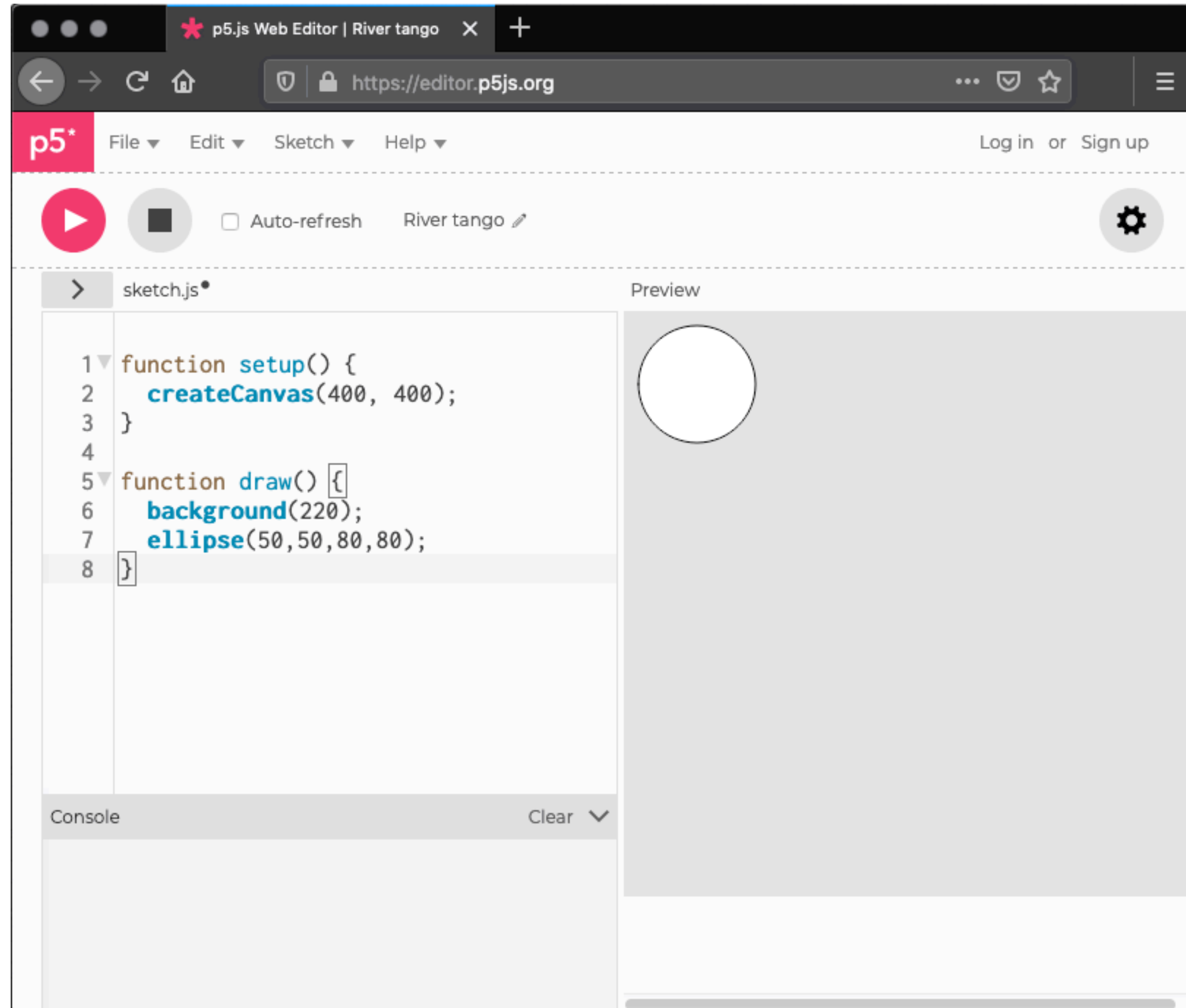
Looking for the old p5.js site? Find it here!



Javascript syntax

setup() called
once at the
beginning

draw() called in a
loop



The screenshot shows the p5.js Web Editor interface. The browser address bar displays `https://editor.p5js.org`. The editor's top navigation bar includes a red 'p5*' logo, a menu with 'File', 'Edit', 'Sketch', and 'Help', and links for 'Log in' or 'Sign up'. Below the navigation bar, there are control buttons: a red play button, a black square, an 'Auto-refresh' checkbox, and the user name 'River tango' with an edit icon. A settings gear icon is on the right. The main workspace is split into two panes: 'sketch.js' on the left and 'Preview' on the right. The code in 'sketch.js' is as follows:

```
1 function setup() {  
2   createCanvas(400, 400);  
3 }  
4  
5 function draw() {  
6   background(220);  
7   ellipse(50, 50, 80, 80);  
8 }
```

The 'Preview' pane shows a white circle on a light gray background. At the bottom, there is a 'Console' pane with a 'Clear' button and a dropdown arrow.

Useful functions

background(color)

Examples

```
Press Shift-Space to insert tab. edit reset copy
// A grayscale integer value.
background(51);
describe('A canvas with a dark charcoal gray background.');
```

```
Press Shift-Space to insert tab. edit reset copy
// A grayscale integer value and an alpha value.
background(51, 0.4);
describe('A canvas with a transparent gray background.');
```

```
Press Shift-Space to insert tab. edit reset copy
// R, G & B integer values.
background(255, 204, 0);
describe('A canvas with a yellow background.');
```

```
Press Shift-Space to insert tab. edit reset copy
// H, S & B integer values.
colorMode(HSB);
background(255, 204, 100);
describe('A canvas with a royal blue background.');
```

color can be

- 1 argument: grayscale value (0-255)
- 2 arguments: grayscale value & opacity (0-1)
- 3 arguments: (red, green, blue) (0-255)
- 3 arguments: (hue, saturation, value)
- 1 argument: hex code '#00ff00'
- 1 argument: CSS named color 'magenta'
- and more!

fill(color)

applies to all shapes after

```
Press Shift-Space to insert tab. edit reset copy
// Six-digit hex RGB notation.
fill('#A251FA');
square(20, 20, 60);
describe('A purple square with a black outline.');
```

what kinds of shapes?

line(x1, y1, x2, y2)

ellipse(x, y, w, [h])

rect(x, y, w, [h])

...

Read the reference docs!!!

[Home](#)[Editor](#)[Download](#)[Donate](#)[Get Started](#)[Reference](#)[Libraries](#)[Learn](#)[Teach](#)[Examples](#)[Contribute](#)[Books](#)[Community](#)[Showcase](#)

Reference

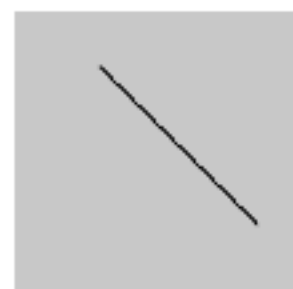
line()

Description

Draws a line, a straight path between two points. Its default width is one pixel. The version of `line()` with four parameters draws the line in 2D. To color a line, use the `stroke()` function. To change its width, use the `strokeWeight()` function. A line can't be filled, so the `fill()` function won't affect the color of a line.

The version of `line()` with six parameters allows the line to be drawn in 3D space. Doing so requires adding the `WEBGL` argument to `createCanvas()`.

Examples



Press Shift-Space to insert tab.

[edit](#) [reset](#) [copy](#)

```
line(30, 20, 85, 75);
describe(
  'A black line on a gray canvas running from
  top-center to bottom-right.'
);
```

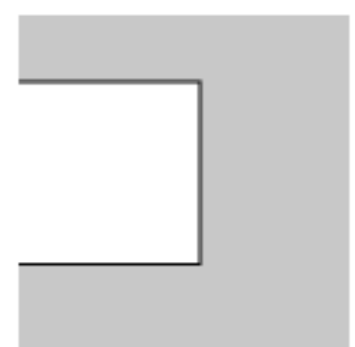
Useful functions

transformations stack & apply to objects after

translate(x, y)

rotate(angle) ...

Examples



Press Shift-Space to insert tab.

edit reset copy

```
translate(0, 20);  
rect(0, 0, 55, 55);
```

translate(20, 0)

translate(50, 0)



translate(70, 0)

same thing

<old settings>

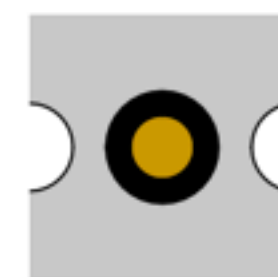
push()

<new settings>

pop()

<old settings>

Examples



Press Shift-Space to insert tab.

edit reset copy

```
ellipse(0, 50, 33, 33); // Left circle
```

```
push(); // Start a new drawing state
```

```
strokeWeight(10);
```

```
fill(204, 153, 0);
```

```
translate(50, 0);
```

```
ellipse(0, 50, 33, 33); // Middle circle
```

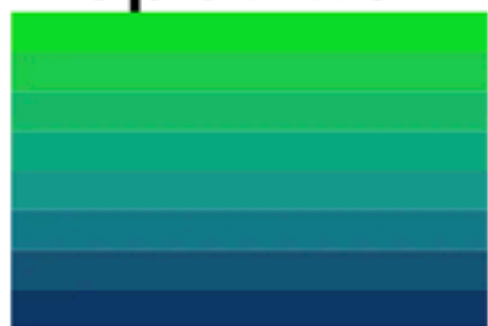
```
pop(); // Restore original state
```

```
ellipse(100, 50, 33, 33); // Right circle
```

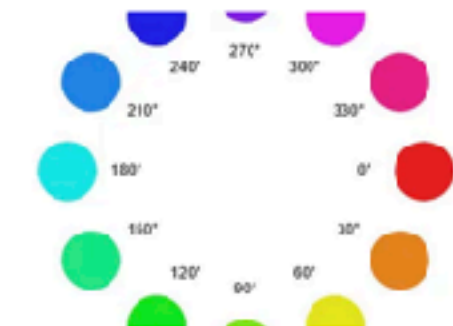


Clock
Get the current time.

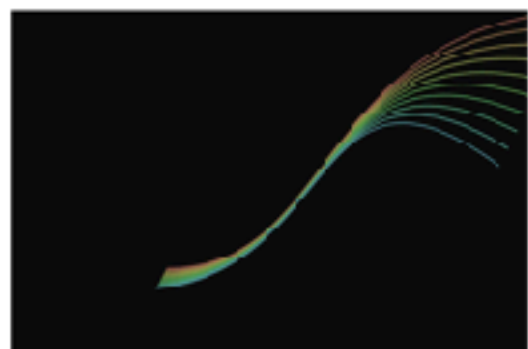
Repetition



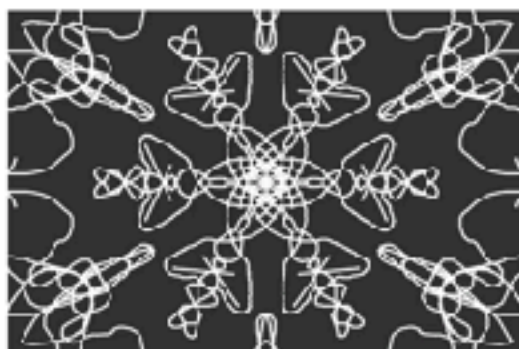
Color Interpolation
Fade between two colors.



Color Wheel
Create a visualization of the color spectrum.



Bezier
Draw a set of curves.



Kaleidoscope
Draw mirrored designs with the mouse.



Noise
Generate naturalistic textures using Perlin noise.



Recursive Tree
Draw a tree using a function that calls itself.

Start Coding

Donate

Jump to

- Shape
- Color
- Typography
- Image
- Transform
- Environment
- 3D
- Rendering
- Math
- IO
- Events
- DOM
- Data
- Structure
- Constants
- Foundation

Reference

Find easy explanations for every piece of p5.js code.

Filter by keyword

Looking for p5.sound? Go to the [p5.sound reference!](#)

Shape

2D Primitives

`arc()`
Draws an arc.

`circle()`
Draws a circle.

`ellipse()`
Draws an ellipse (oval).

`line()`
Draws a straight line between two points.

`point()`
Draws a single point in space.

`quad()`
Draws a quadrilateral (four-sided shape).

`rect()`
Draws a rectangle.

`square()`
Draws a square.

`triangle()`
Draws a triangle.

Start Coding

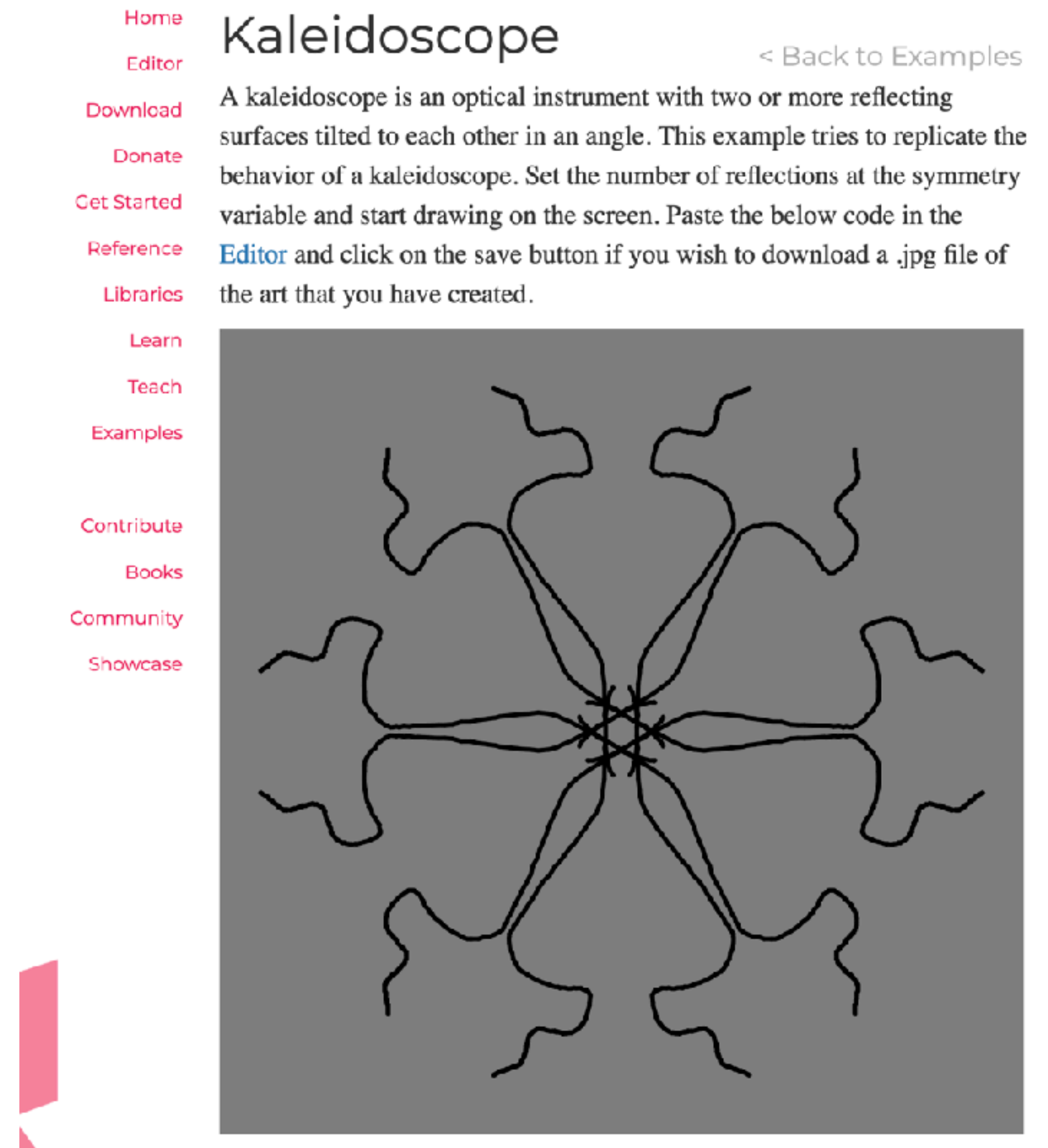
Donate

Jump to

- Featured
- Shapes And Color
- Animation And Variables
- Imported Media
- Input Elements
- Transformation
- Calculating Values
- Repetition
- Listing Data with Arrays
- Angles And Motion
- Games
- 3D
- Advanced Canvas

Your task: edit Kaleidoscope example

- Examples (sidebar) -> Repetition -> Kaleidoscope
- Paste example code into editor
- Make at least **3 meaningful lines of code changes** resulting in a **visually different** piece
- Learning goal: experiment with p5.js, just change random variables and see what happens, understand creative coding process
- Save and upload drawings on Canvas (p5.js art gallery assignment)

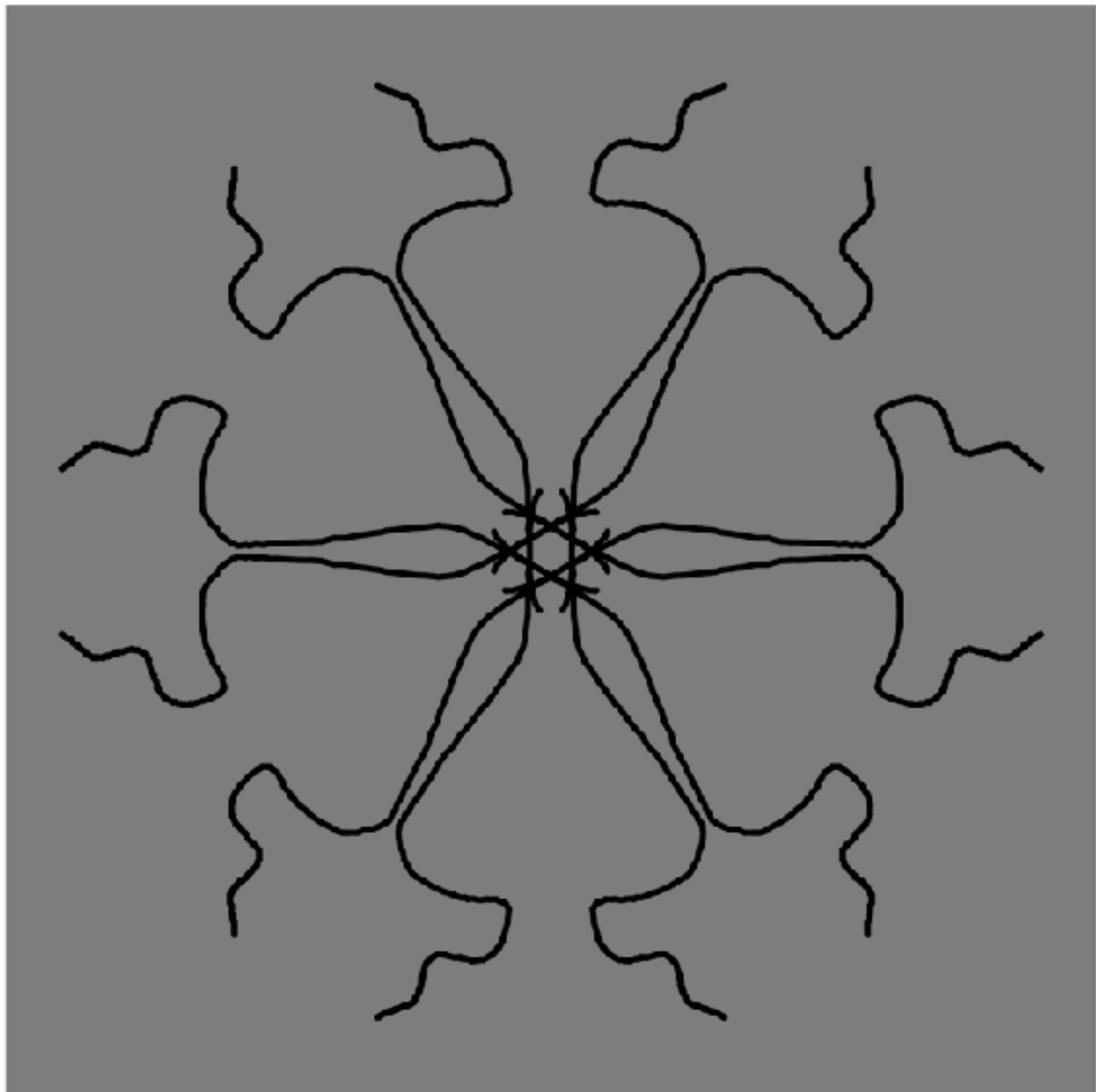


Home
Editor
Download
Donate
Get Started
Reference
Libraries
Learn
Teach
Examples
Contribute
Books
Community
Showcase

Kaleidoscope

[< Back to Examples](#)

A kaleidoscope is an optical instrument with two or more reflecting surfaces tilted to each other in an angle. This example tries to replicate the behavior of a kaleidoscope. Set the number of reflections at the symmetry variable and start drawing on the screen. Paste the below code in the [Editor](#) and click on the save button if you wish to download a .jpg file of the art that you have created.



Instructions

- First, no coding: browse the Reference and Tutorial links to understand the p5.js API and brainstorm changes you can make (~ 5-10 min)
- Pair program to change the kaleidoscope example
- Make at least **3 meaningful lines of code changes** resulting in a **visually different** piece
- Save and upload drawings on Canvas (p5.js art gallery assignment)
- If you're done early, type a short reflection: How did this experience differ from other kinds of coding? What was challenging about the process of being expressive?

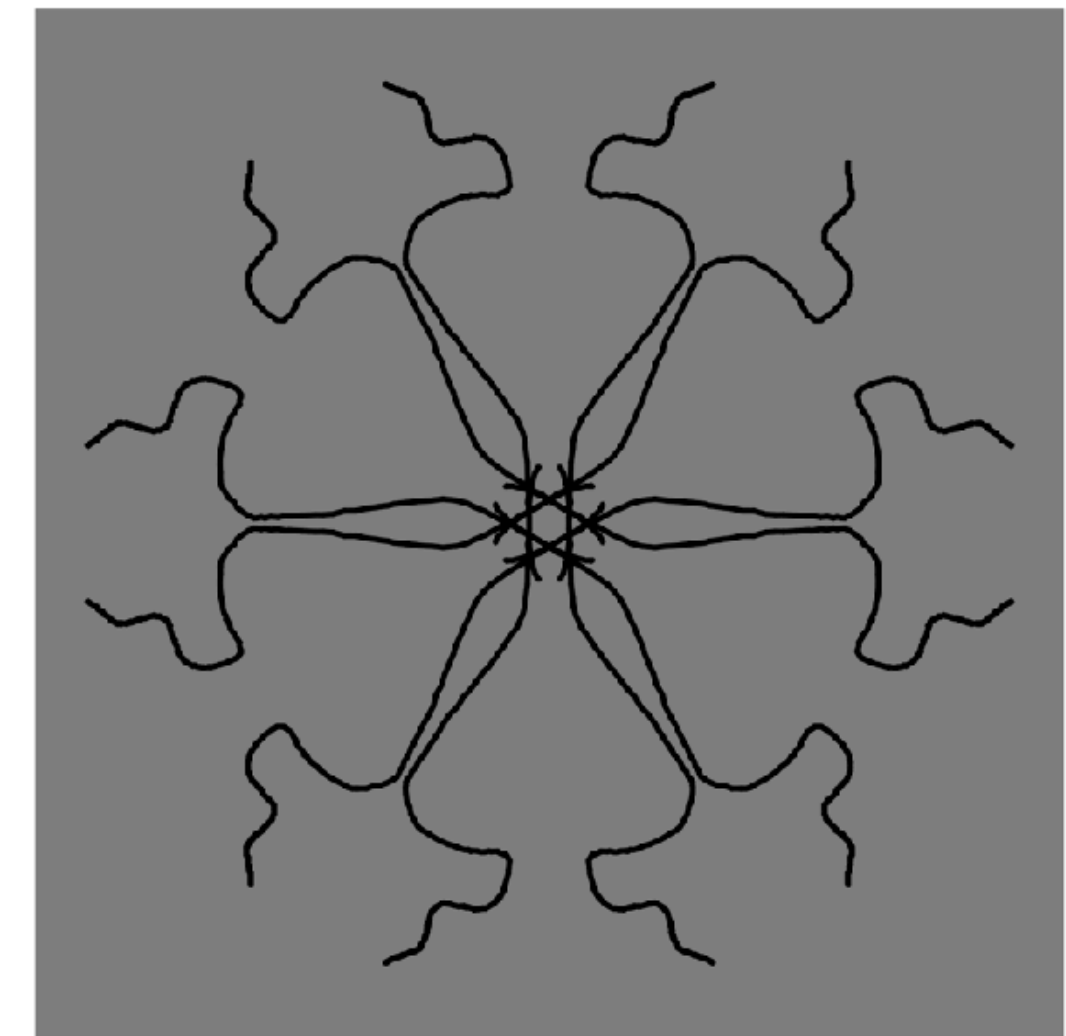
p5.js

Home
Editor
Download
Donate
Get Started
Reference
Libraries
Learn
Teach
Examples
Contribute
Books
Community
Showcase

Kaleidoscope

[< Back to Examples](#)

A kaleidoscope is an optical instrument with two or more reflecting surfaces tilted to each other in an angle. This example tries to replicate the behavior of a kaleidoscope. Set the number of reflections at the symmetry variable and start drawing on the screen. Paste the below code in the [Editor](#) and click on the save button if you wish to download a .jpg file of the art that you have created.



This assignment does not count toward the final grade.

p5.js art gallery

Published

Edit

⋮

Upload your image here (one person per group is fine).

Points 0

Submitting a file upload



How does your experience of "creative coding" differ from your usual CS coding experience? You may talk about process, expressivity, or anything else.

Join at
slido.com
#1819

Class 9 recap

- TODOs:
 - By **Wednesday's** class:
 - RRs x 2 (remember, you can drop 4!)
 - Recommended deadline for PM4's storyboard
 - Next week - pre-fall break push!!
 - **Monday:** PM4 (3D printing for protest)
 - Weds: Project group formation