

Creative Coding I

MA Creative Technologies - WS '25'26



FILMUNIVERSITÄT
BABELSBERG
KONRAD WOLF

Jonathan Ho

Creative Technologies

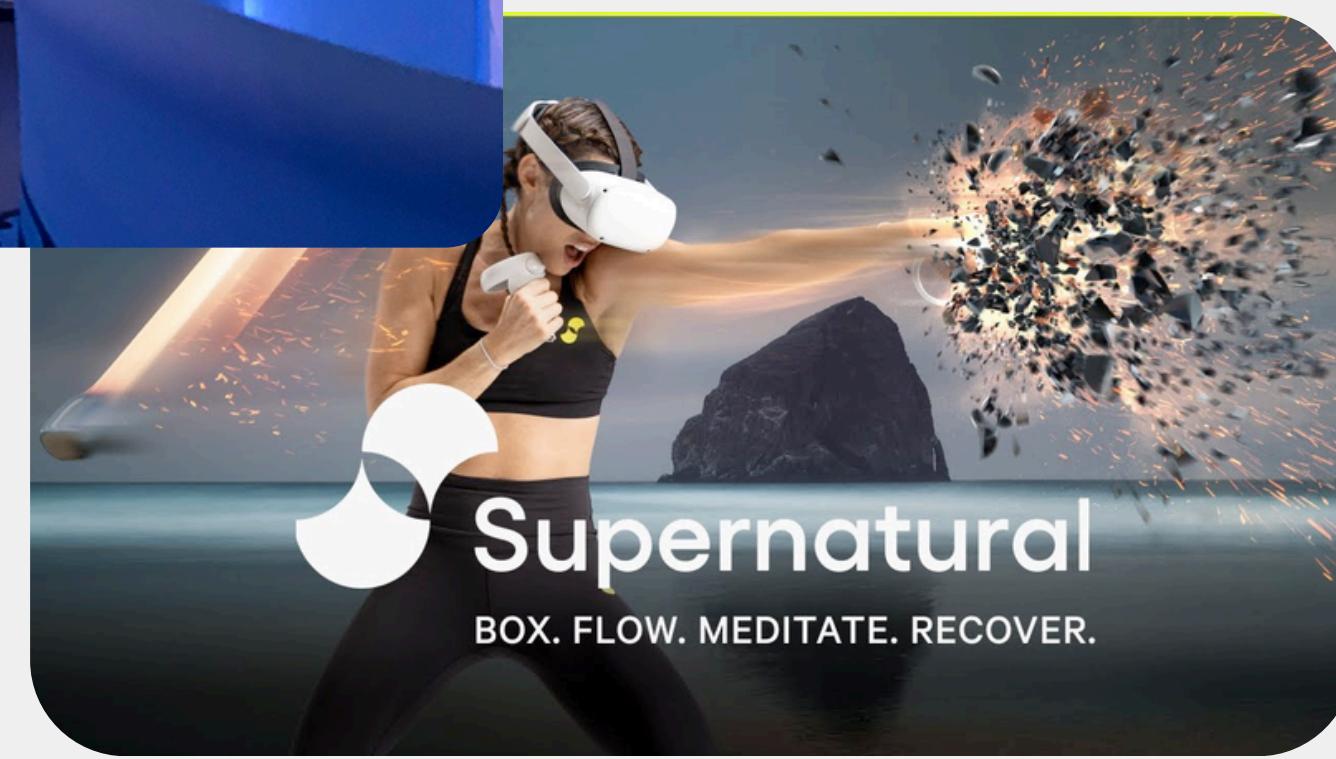
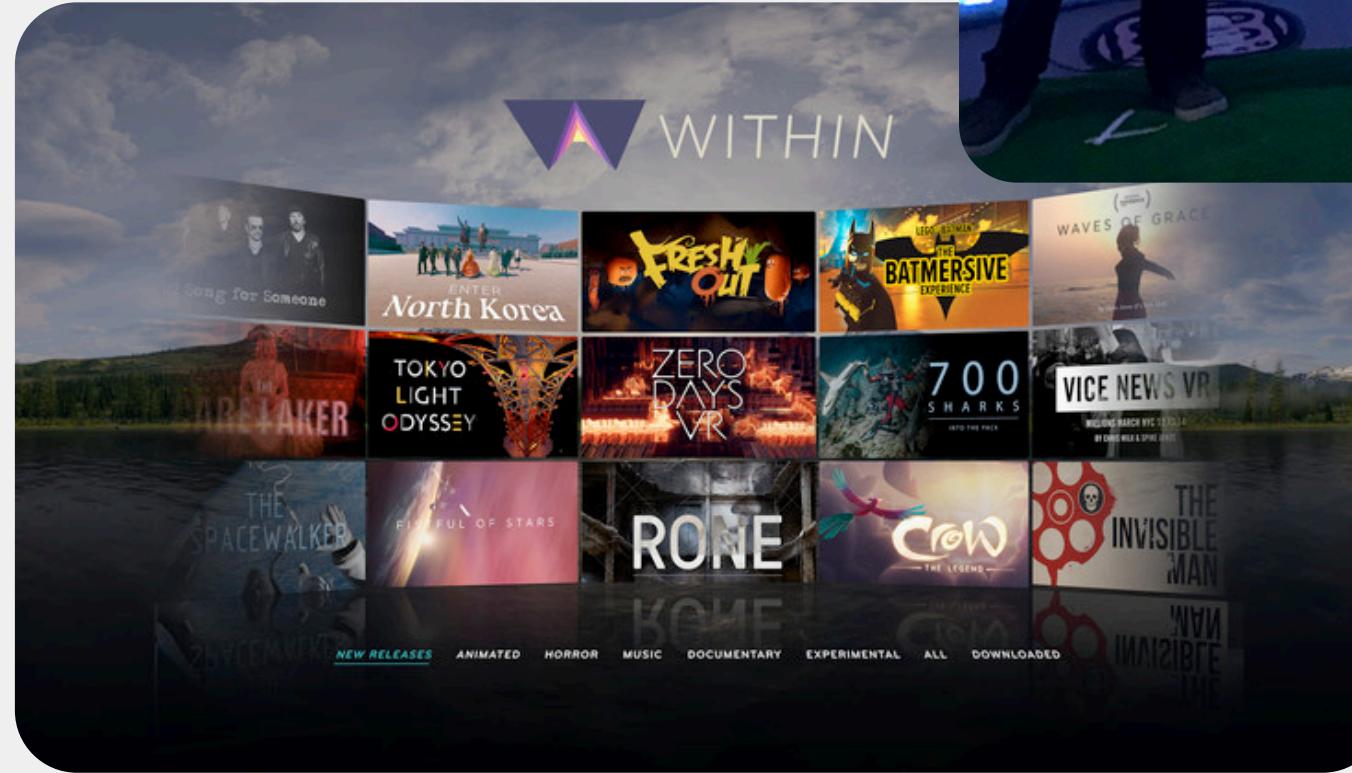
A Bit About Me

From Film sound and Visual Effects...



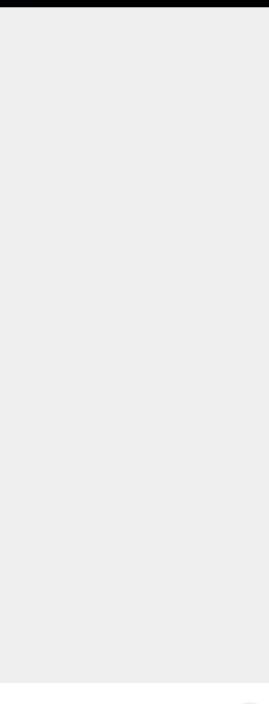
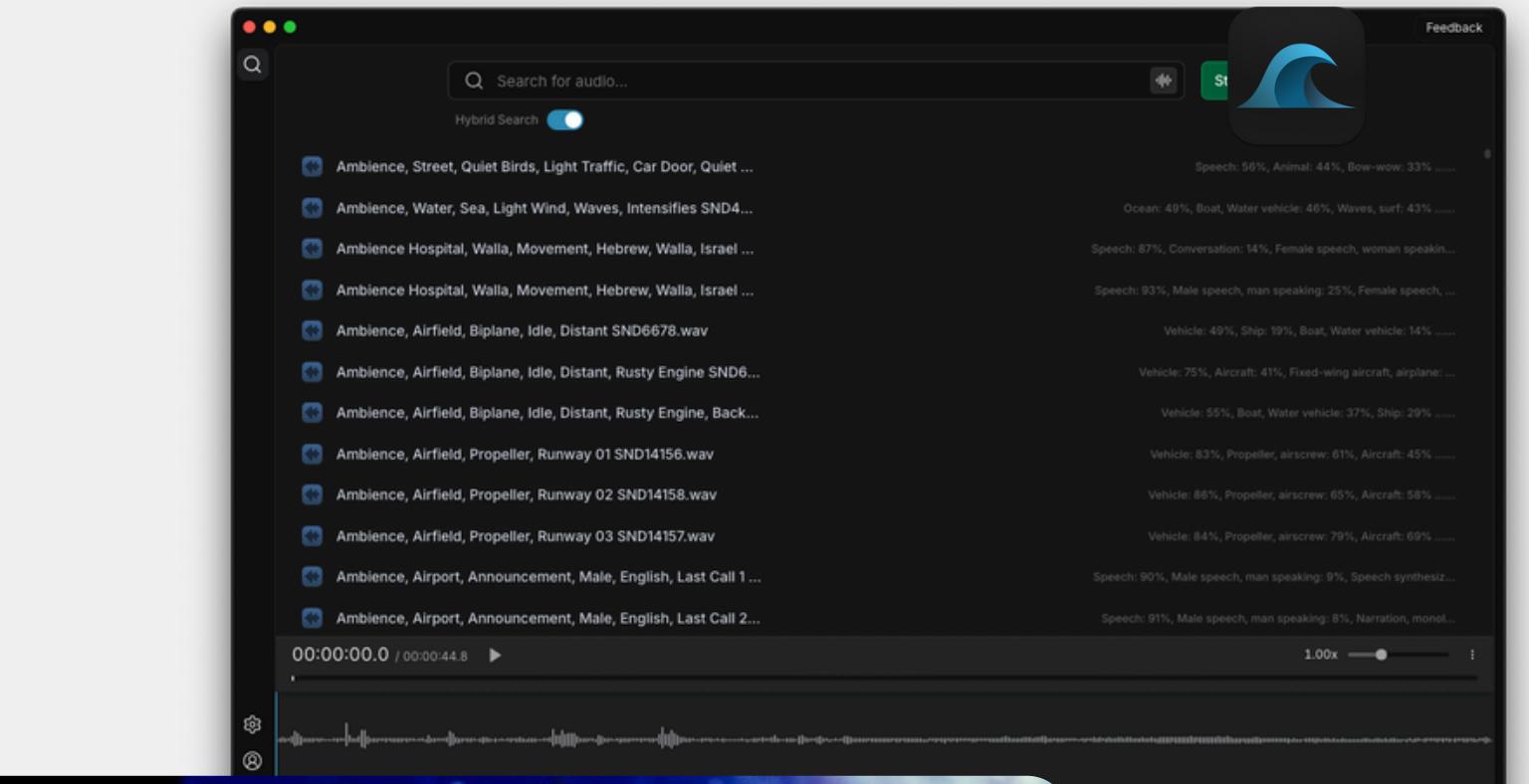
Into Virtual Reality...

at WITHIN in LA



...to CTech

Started MA in 2021



How do you define Creativity?

In what ways do you express creativity?

What do you understand under the term coding?

How do you code?

Today

01 Creative Coding

What is it?

02 Examples

What have others done?

03 Learning Objective

Coding in this class

Conceptualization and design

04 Housekeeping things

README.md

05 Emergence

05 p5.js

01.

Creative Coding

"In some ways, **programming** is like **painting**. You start with a blank canvas... You use a combination of science, art, and craft to determine what to do with them."

- Andrew Hunt (*author, the pragmatic programmer*)

Question:

What could we consider practices of creative coding?

Producing something expressive rather than practical

Programming for the sake of art

Building tools that help others be creative

Giving others the ability to create or collaborate

Using software or hardware in unintended ways

Bending or hacking software or hardware creatively

??

Aesthetics, insight, joy, dialog, politics, collaboration, augmentation, emotion, perspectives, friendship,...

How could you explore one of the above mentioned terms with a software project?

Creative Coding

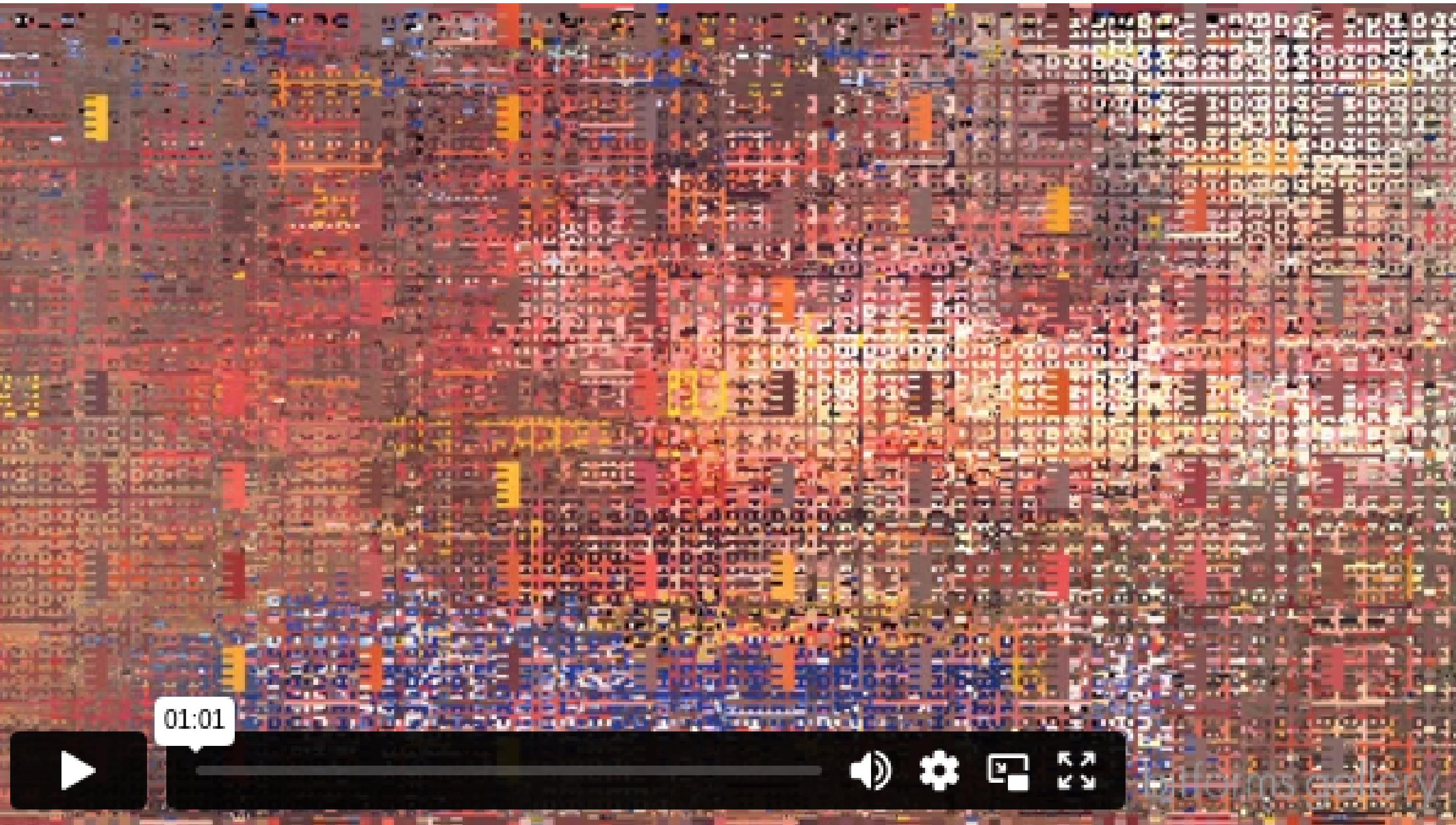
A creative approach to coding

**What do I have available, and what
can I do with it beyond the obvious?**



Random International - Audience (2008)

Mounted on small metal feet, the horde of mirrors are clearly mechanical yet the demeanour of their movement gives a sense of abstracted, human-like behaviour making them appear oddly anthropomorphic. Through simple pan and tilt gestures akin to turning one's head, the mirrors convey individual characters but when a person approaches, they all turn to face the visitor in one inquisitive, synchronised gesture. Reflected in every mirror, the onlooker becomes the subject of their own gaze and that of the artwork.



Infinite Command Team by Casey Reas

Casey Reas' Infinite Command Team investigates the relationship between particles that are encoded to construct images, and the code that forges those particles. Using pixelation of different weights and sizes, the piece creates a digital mosaic of television signals that become abstract and collage-like, reminiscent of TV channel-surfing. The piece is a celebration of art and technology that showcases the potential of combining digital fragments into a holistic piece of work.



Watch video on YouTube

Error 153

Video player configuration error



16



Unnumbered Sparks - Janet Echelman & Aaron Koblin (2014)

A monumental interactive sculpture in the sky. Choreographed by visitors in real time through their mobile devices, the sculpture is a crowd-controlled visual artwork on a giant, floating canvas. A collaboration with Janet Echelman premiered at the TED conference in 2014.



The Jonny Cash Project Chris Milk & Aaron Koblin

A hand-drawn, animated, music video with each frame drawn by a different person. Participants are invited to create a drawing that is woven into a collective music-video-tribute to Johnny Cash. Set to his song "Ain't No Grave," the project was inspired Johnny's central lyric "Ain't no grave can hold my body down." It represents Cash's continued existence, even after his death, through his music and his fans. The work continues to grow and evolve as more people participate.

A collaboration with director [Chris Milk](#).

Everything in Existence is the first solo exhibition by fuse* and traces a line that highlights the evolution of the studio's practice over the past ten years of activity. Presenting four multimedia installations that invite audiences to experience different perceptions of reality, the exhibition is designed to remind us that we are all part of something bigger. The show is inspired by the concept of interconnectedness, the belief that we are all part of everything in existence.

The works exhibited share common ground since they are generated by software processing data in real-time. The data may be derived from interactions with the viewer (Snowfall), from social networks (Amygdala), from sound (Clepsydra) or from the software itself (Multiverse). This generative technique creates "living" art that constantly renews itself and changes before one's eyes, rewarding prolonged viewing and repeat visits from the spectator. This interplay between the human hand of the artist or the participant and the invisible inner workings of the algorithmic software, which ultimately creates the artworks, parallels the relationship between humans, in all of their vulnerability, and the mysterious, extremely vast and incomprehensible forces that push us towards the unknown, beyond the limits of the known world. [\[project website\]](#)



Everything in Existence by fuse*

19

About the project

The image is a composite of several screenshots from a video conference and a digital workspace.

Video Conference Participants:

- Top-left: A woman with glasses and a white shirt.
- Top-right: Two women smiling and waving.
- Middle-left: A man with glasses and a beard, and a woman with dark hair.
- Middle-right: A woman with dark hair and headphones.
- Bottom-left: A man with glasses and a beard.
- Bottom-right: A woman with headphones.

Miro Board:

A flowchart titled "My First Board" illustrating a process:

```
graph LR; A[Brainstorming session] --> B[New product idea]; B --> C{Prototyping}; C --> D[Prototype evolution]; D --> E[Review]; E --> F[Quick design]; F --> G[Brainstorming session];
```

Below the flowchart is a Kanban board with columns: Application setup, Transactions, Address city, Interactions with application, Settings, Finance management, Bank products, Bank services, Analytics, and Data visualization.

Dashboard:

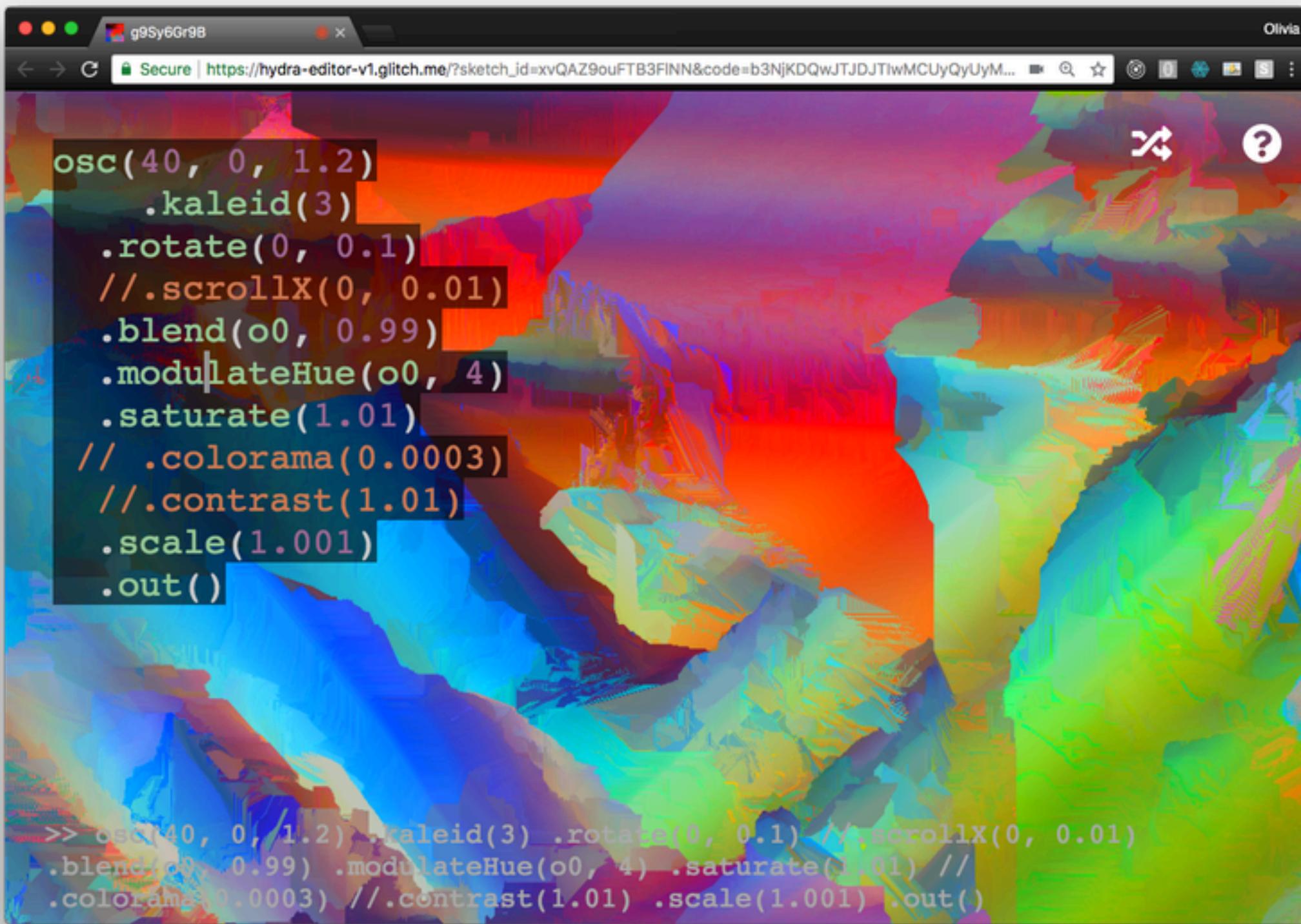
A dashboard titled "Nov'16. Traffic sources, Similarweb" showing a pie chart of traffic sources:

Traffic Source	Percentage		
Direct 51.1%	Search 38.7%	Referral 14.2%	Social 3.1%
Search 38.7%	Referral 14.2%	Social 3.1%	
Referral 14.2%	Social 3.1%		
Social 3.1%			

Landing Page:

A landing page for a company with the tagline "We are on a mission to help teams create the Next Big Thing". It features sections for "The world of work has dramatically changed", "We help teams and organizations", "We follow three values building our business, team and brand", and "Play as a team, to win the world".

Zoom & Miro

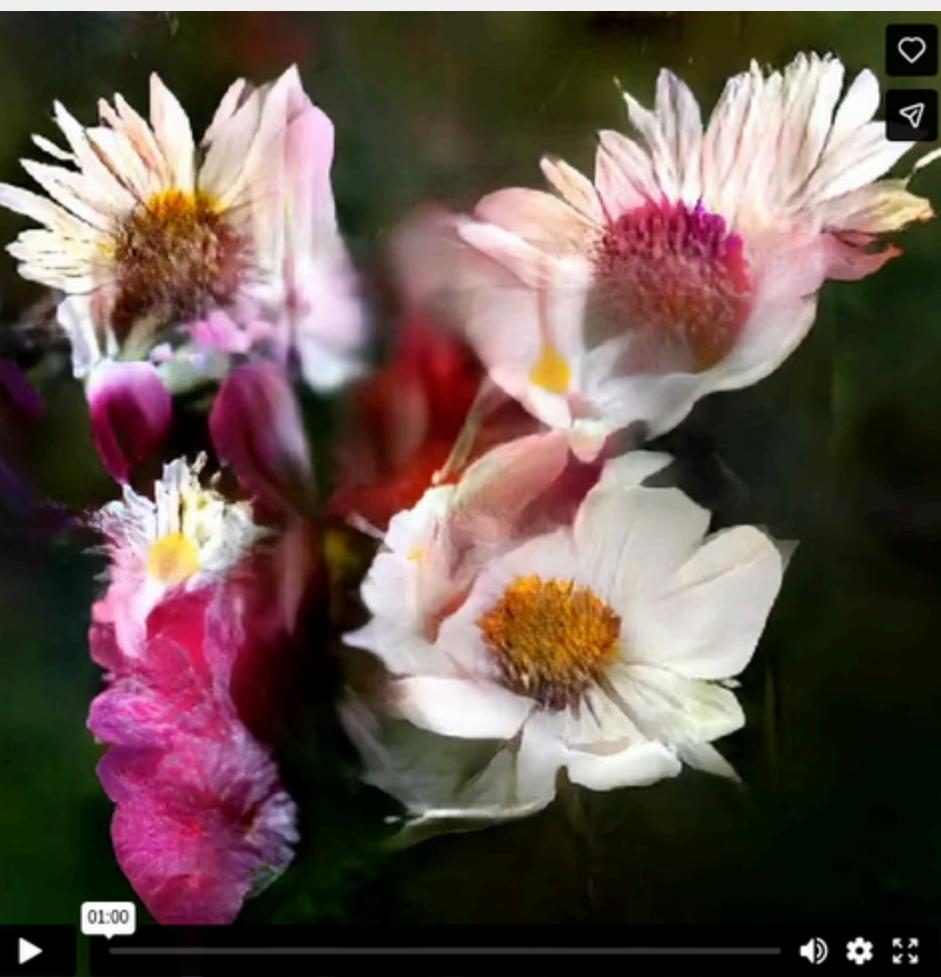


- For audivisual performances
- Open-source
- All-levels

Learning to See is an ongoing collection of works that use machine learning algorithms to reflect on ourselves and how we make sense of the world. The picture we see in our conscious mind is not a mirror image of the outside world, but is a reconstruction based on our expectations and prior beliefs.

An artificial neural network looks out onto the world, and tries to make sense of what it is seeing. But it can only see through the filter of what it already knows.

In this context, the term seeing, refers to both the low level perceptual and phenomenological experience of vision, as well as the higher level cognitive act of making meaning, and constructing what we consider to be truth. Our self affirming cognitive biases and prejudices, define what we see, and how we interact with each other as a result, fuelling our inability to see the world from each others' point of view, driving social and political polarization. The interesting question isn't only "when you and I look at the same image, do we see the same colors and shapes", but also "when you and I read the same article, do we see the same story and perspectives?".[\[project website\]](#)



Learning to See by Memo Akten

Can Generative AI be considered Creative Coding?

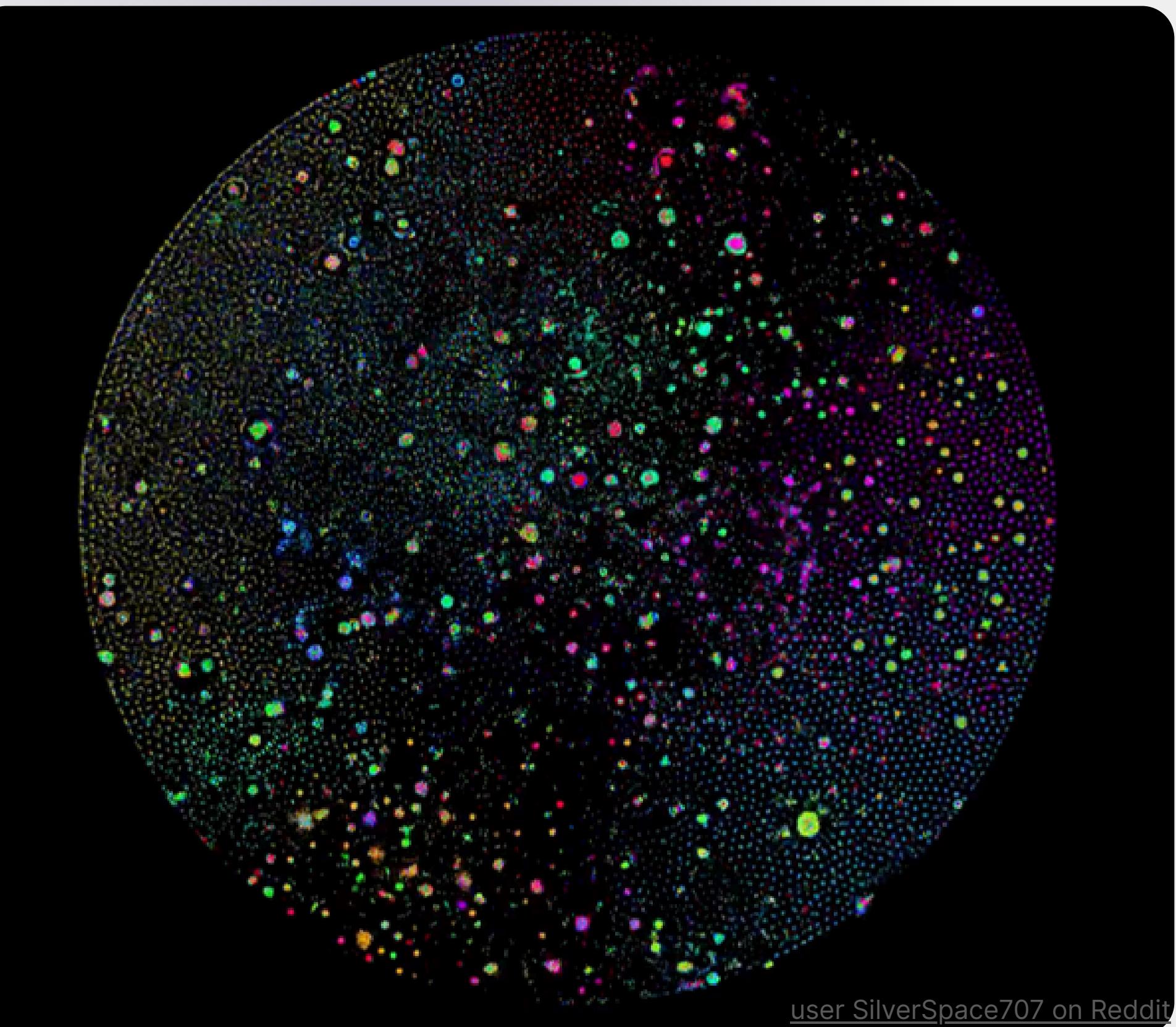
03.

Learning Objectives

What do you want to get from the class?

Learning Objectives

- 01** Advance your ability to write code,
- 02** advance your ability to conceptualize and design,
- 03** acquire fundamental knowledge of web technologies,
- 04** use software development in a creative context, and
- 05** implement specific design goals



[user SilverSpace707 on Reddit](#)

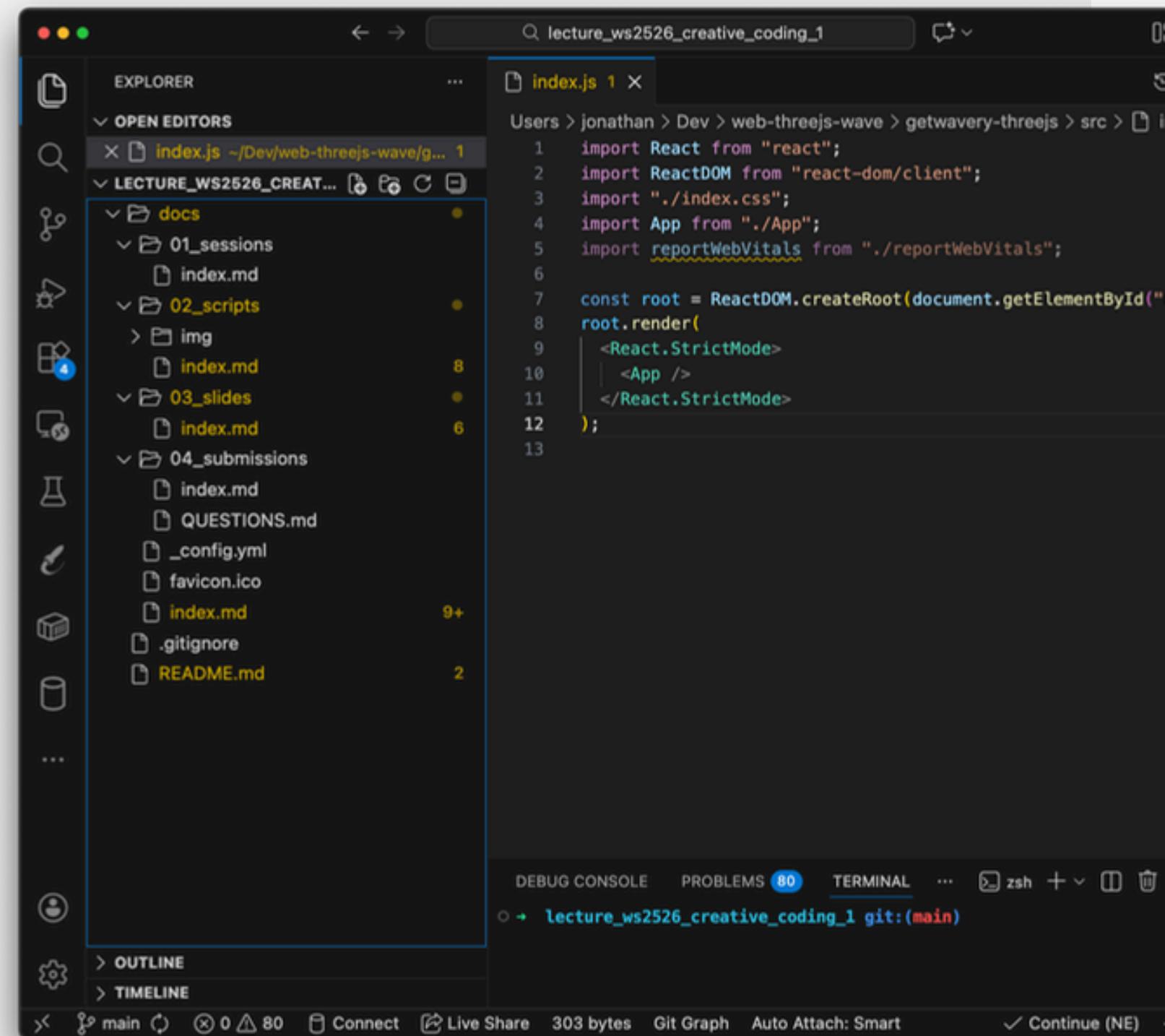
Programming in this Class

Goal

- The focus of this class is not "learn how to program"
- however it is the goal that you all learn how to program with this class. :)

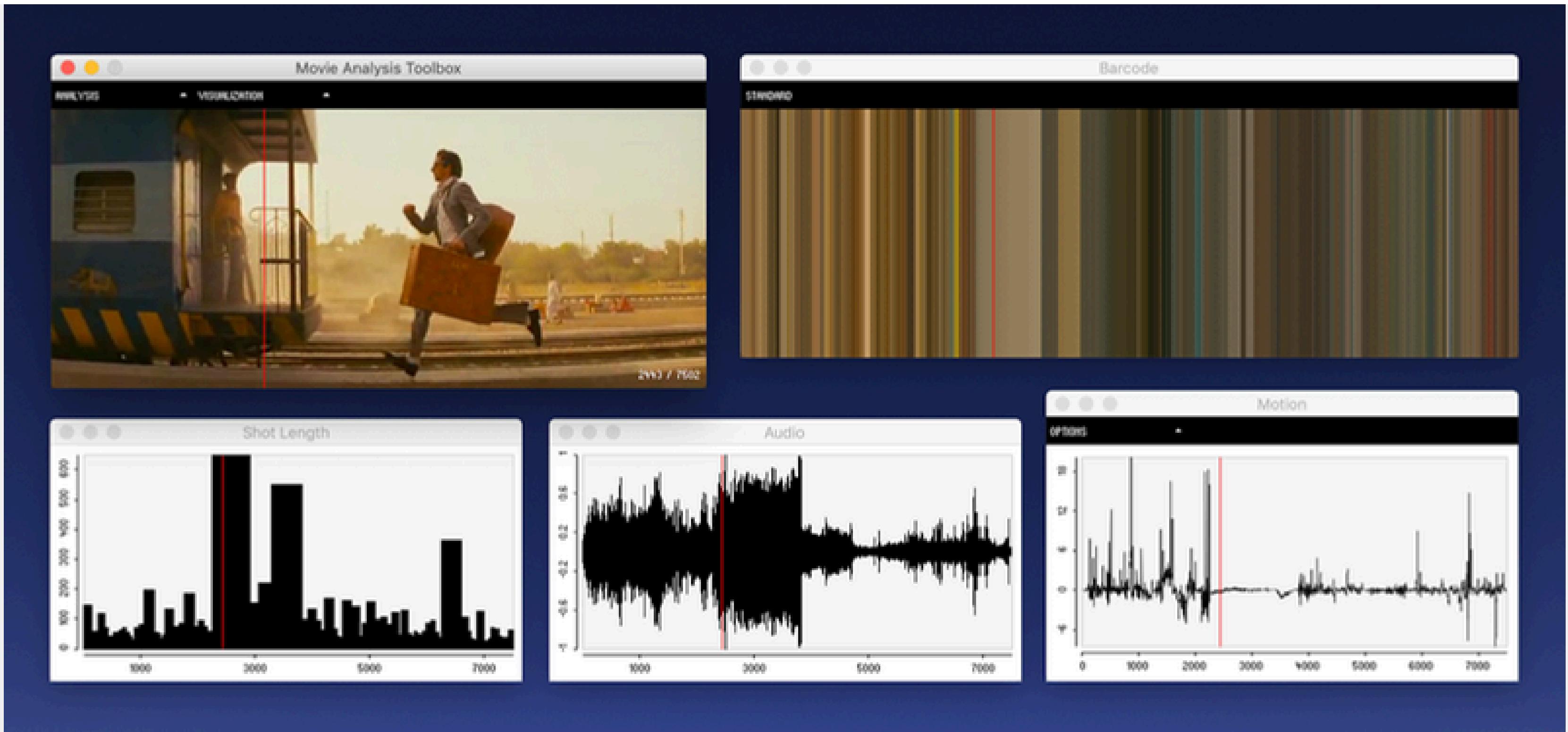
Approach

- Weekly programming assignments
- Assignments are open-ended and are related to the session's topics
- Assignments are evaluated on three levels
 - Concept
 - Form
 - Implementation



A screenshot of a code editor (VS Code) showing a file structure and code. The file tree on the left shows a folder named 'LECTURE_WS2526_CREAT...' containing 'docs' (with '01_sessions', '02_scripts', '03_slides', and '04_submissions' subfolders), 'index.js', 'index.css', 'App.js', and 'reportWebVitals.js'. The 'index.js' file is open in the editor, displaying React code for rendering an App component. The bottom status bar shows the file is 303 bytes and has 80 problems. The bottom right corner shows a small circular icon with the number 26.

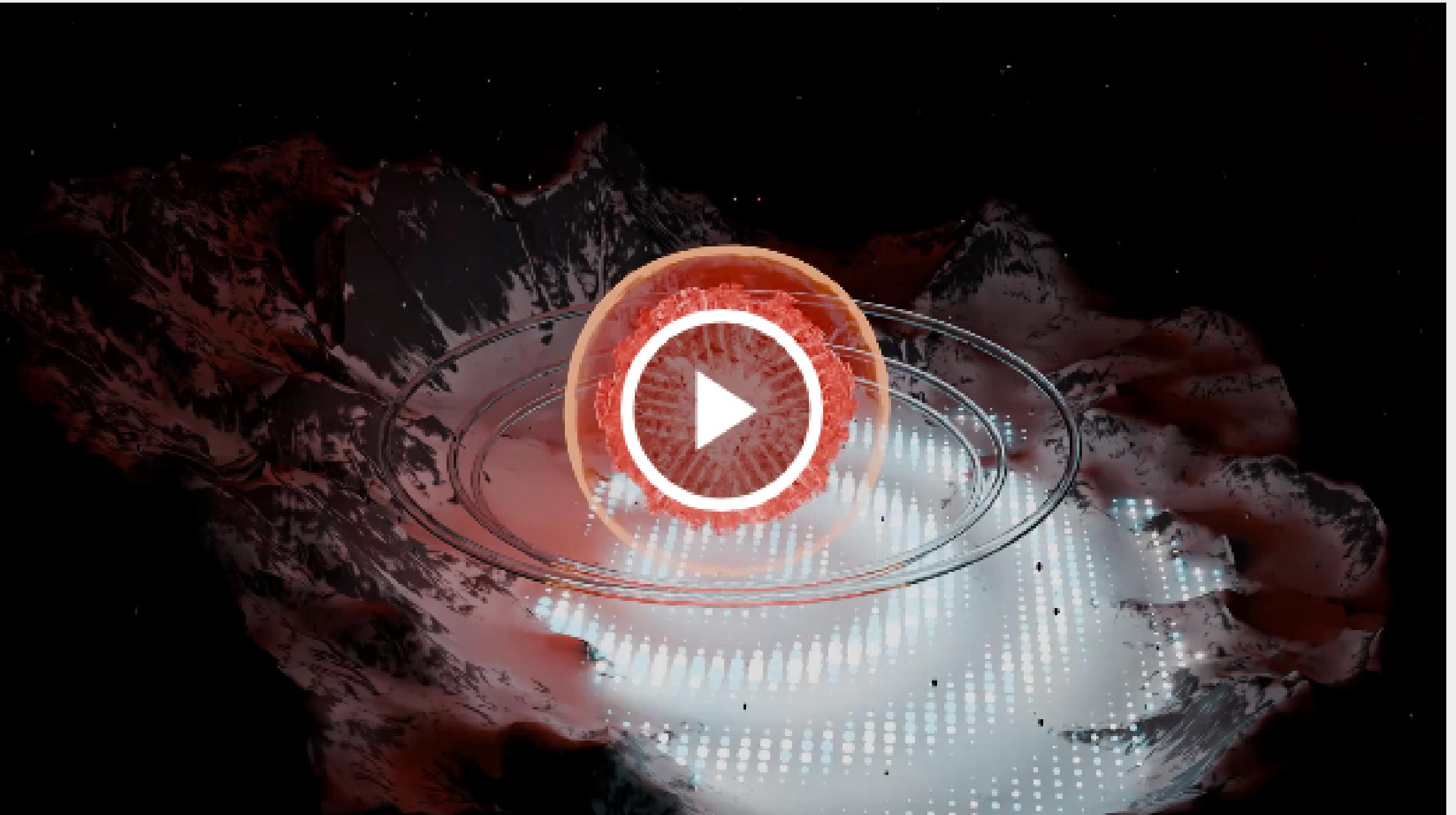
```
Users > jonathan > Dev > web-threejs-wave > getwavery-threejs > src > index.js
1 import React from "react";
2 import ReactDOM from "react-dom/client";
3 import "./index.css";
4 import App from "./App";
5 import reportWebVitals from "./reportWebVitals";
6
7 const root = ReactDOM.createRoot(document.getElementById("root"));
8 root.render(
9   <React.StrictMode>
10     <App />
11   </React.StrictMode>
12 );
```



[by Phil Clausen]

"THERE IS NO TIME" is a project that aims to create a cinematic, mysterious and escapist atmosphere. The cold, harsh environment of an arctic landscape, which at the same time represents a surreal beauty, is intended to create an unnatural, dream-like setting that has a certain hypnotic attraction. However, you shouldn't stay there too long, because the real world and its problems could catch up with you and time is running out. You have to act before it's too late.

Reaper, Blender and Geometry Nodes



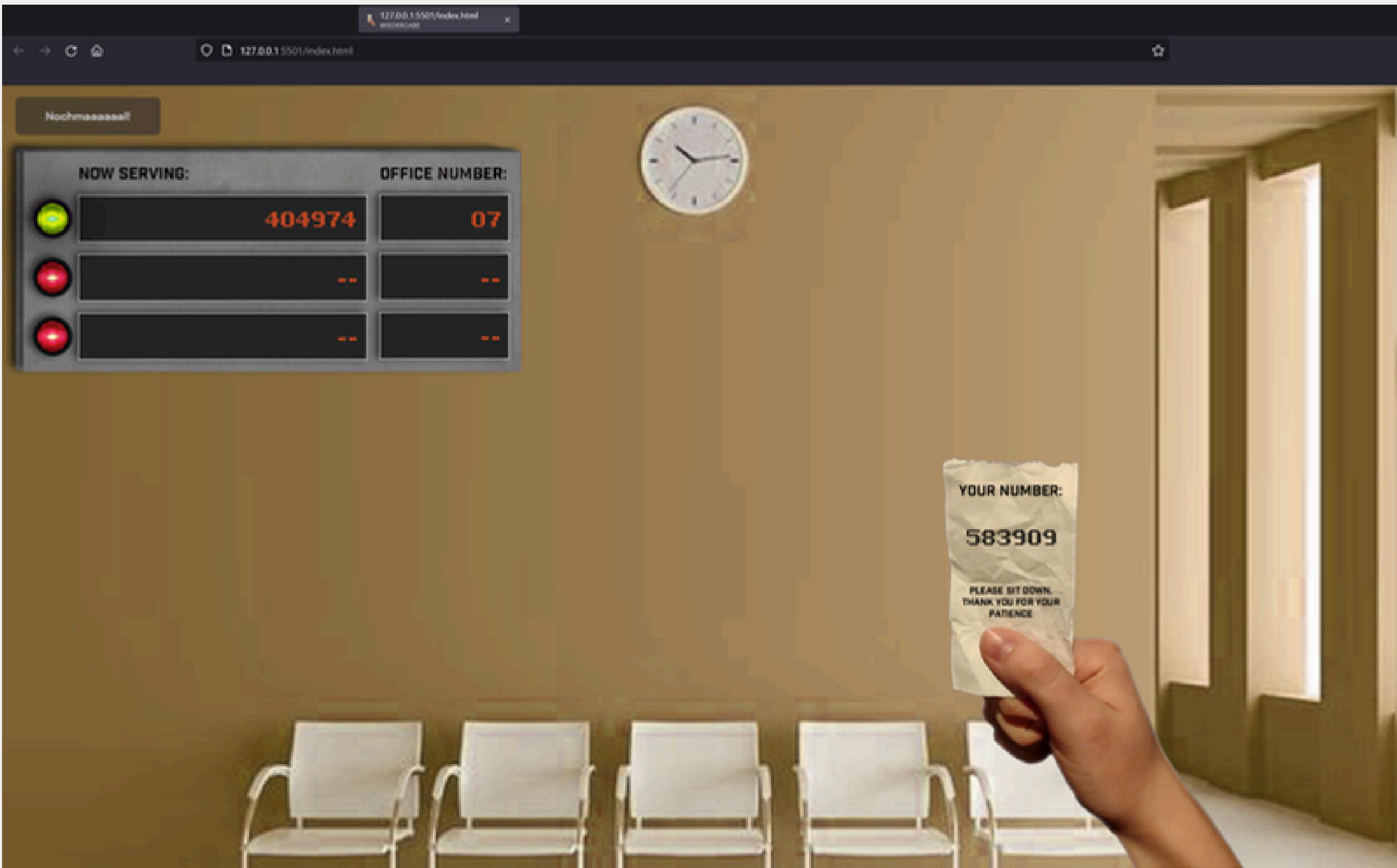
["THERE IS NO TIME" by Maximilian Rueth](#)

The Dark Side of Rock and Roll

In this project, I explore the theory of live decoding audio signals through the creation of a dynamic three js scene. In the set of a dark cave a shiny chromesphere follows the rhythm and character of a dramatic and percussive experimental music track. Sound and motion are synchronized through different volumetric fields of influence. Within certain spectrums of frequency non-trivial systems are pushing and pulling floating rocks around and create on louder sounds captivating impulses.



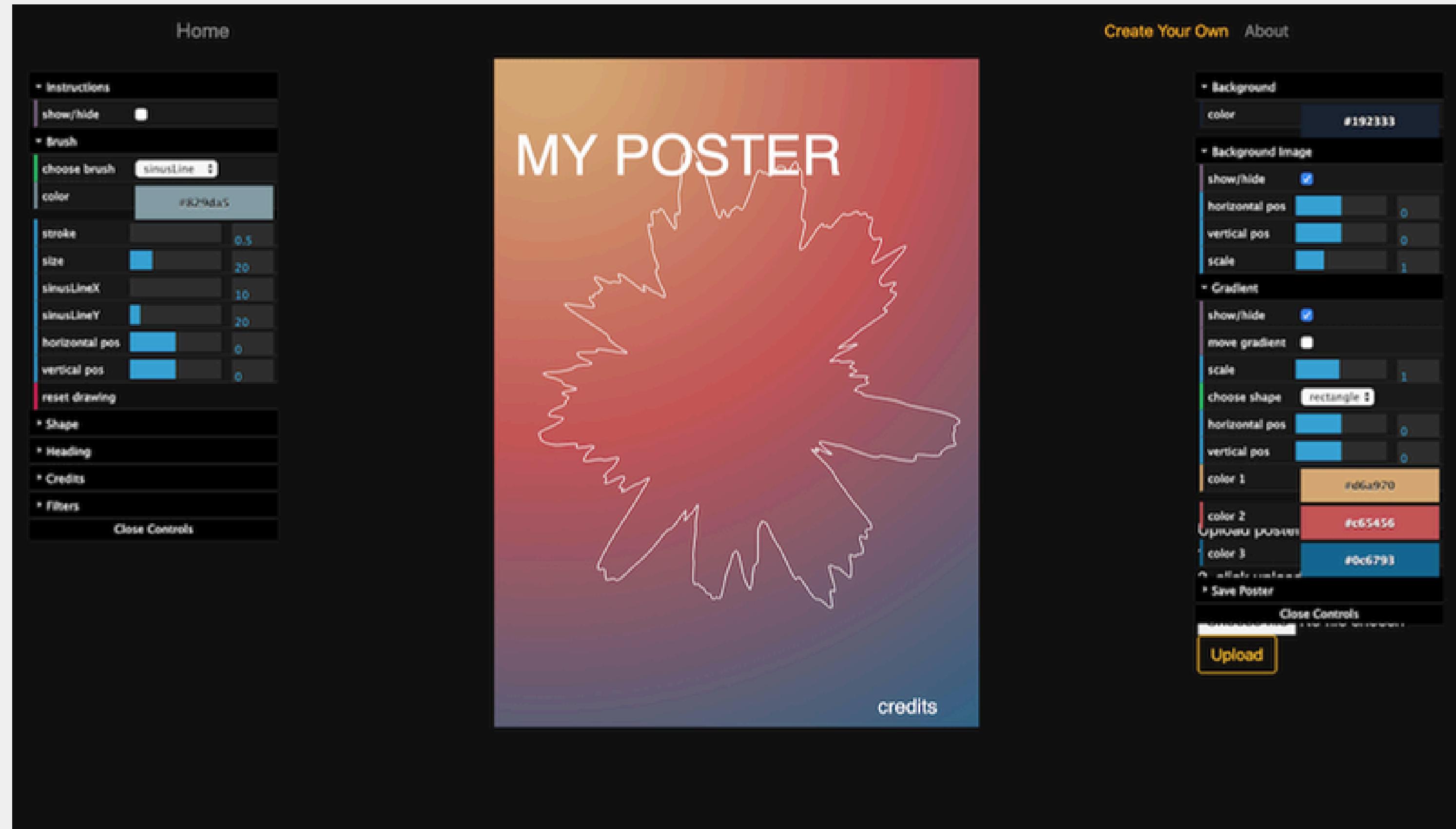
[The Dark Side of Rock and Roll by Adam Streicher](#)



[by Vi Schreiber]

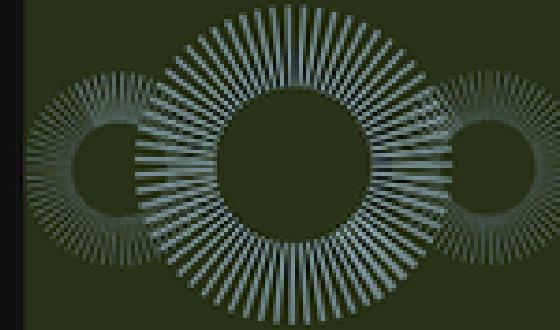


[by Denise Bischof]



[by Anna Eschenbacher & Zainab Tariq]

VIRTUAL PRODUCTION.



"Virtual production is a broad term referring to a spectrum of virtual design, content, and production, and visualization in the working methods of creative industries, primarily motion picture and television production. A common theme is that virtual production is used to reduce the need for shooting on location or in studio sets, to quickly visualize effects in the production schedule, and to cost down production costs. High-quality imagery can be produced from the source, instead of different cameras shooting incompatible video about all their own scenes, which are then composited at a later time for visualization through post-compositing."

this is a rainy poster



creative coding is a type of computer programming in which the goal is to create something expressive instead of something functional.

WITHOUT GRADIENT



without

IT'S AN ISOLATIONSHIP



B

Isolating Blue

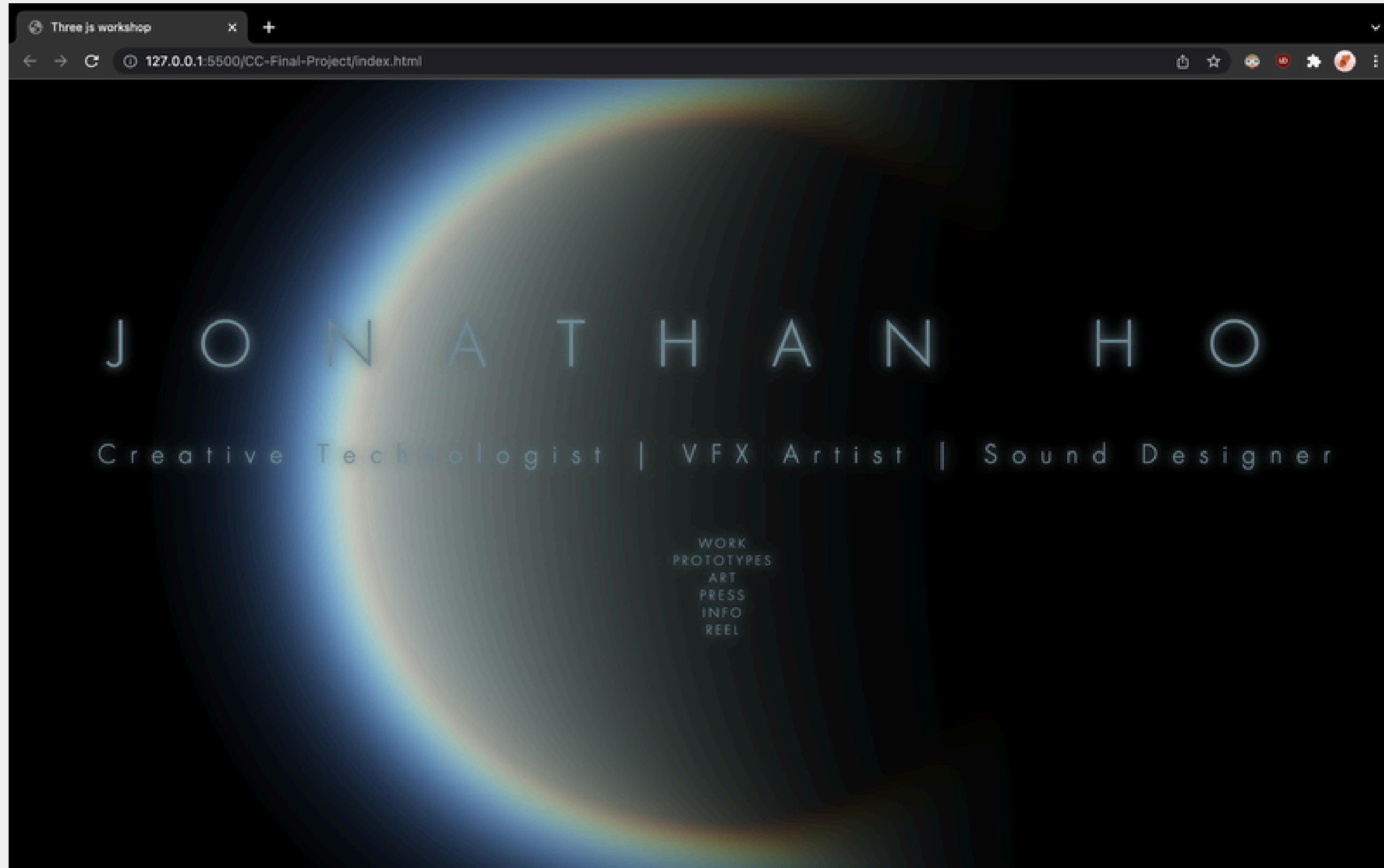


*good booties stay at
home.*

zugfahrt

zugfahrt

[by Anna Eschenbacher & Zainab Tariq]



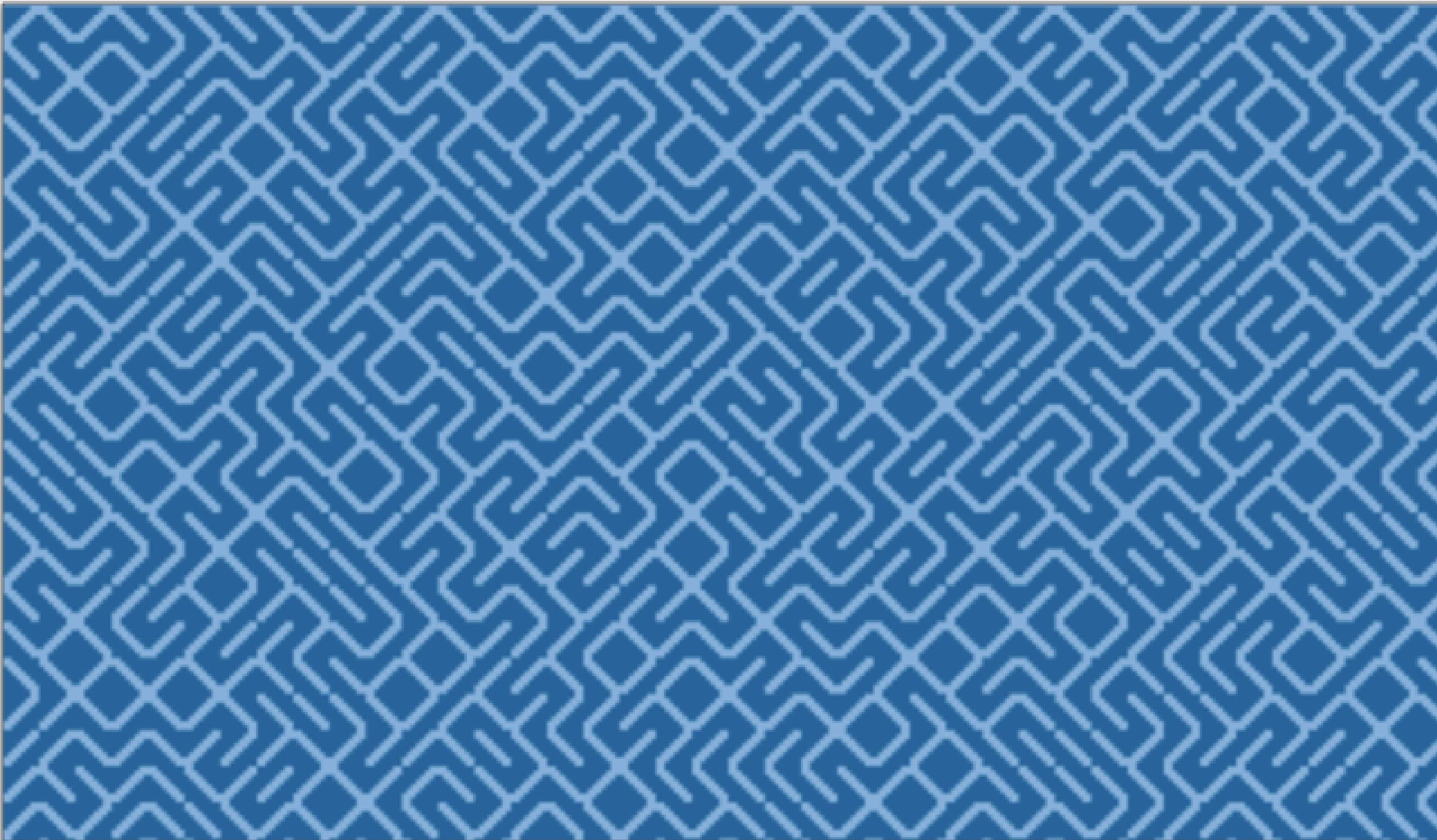
[by me]

Administrative / Housekeeping

README.md

The course syllabus for CC1

Emergence





The original BASIC program for the Commodore 64:

```
10 PRINT CHR$(205.5+RND(1)); : GOTO 10
```

Bash Version: `f="\\\";while :;do print -n ${f[(RANDOM % 2) + 1]};done`

10 Print

- Considered a phenomenon of creative coding through its simplicity and visual appeal
- Additional Reading:
 - There is a dedicated [book](#)
 - Representing the history of creative coding
 - Author talk on [youtube](#)



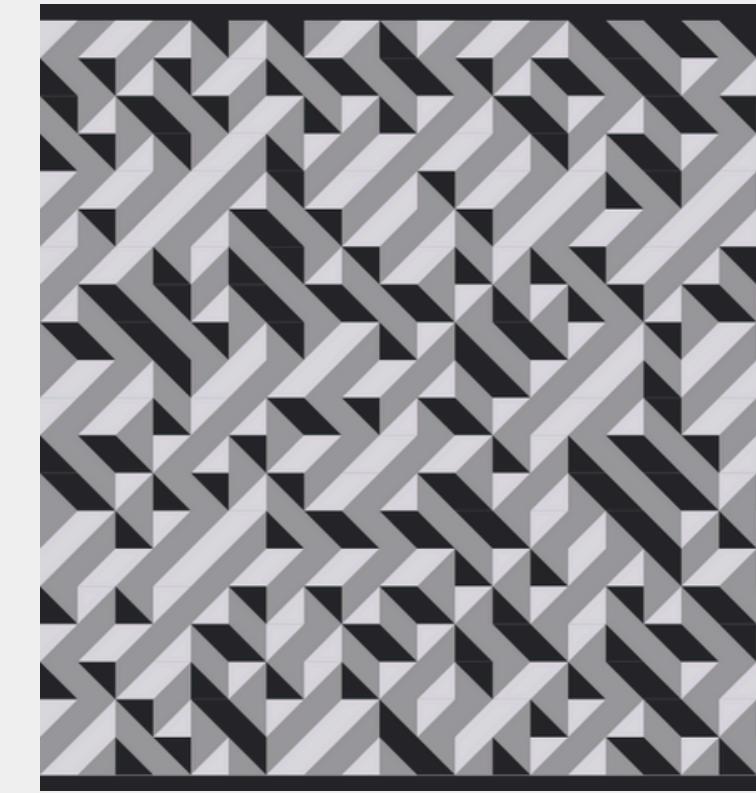
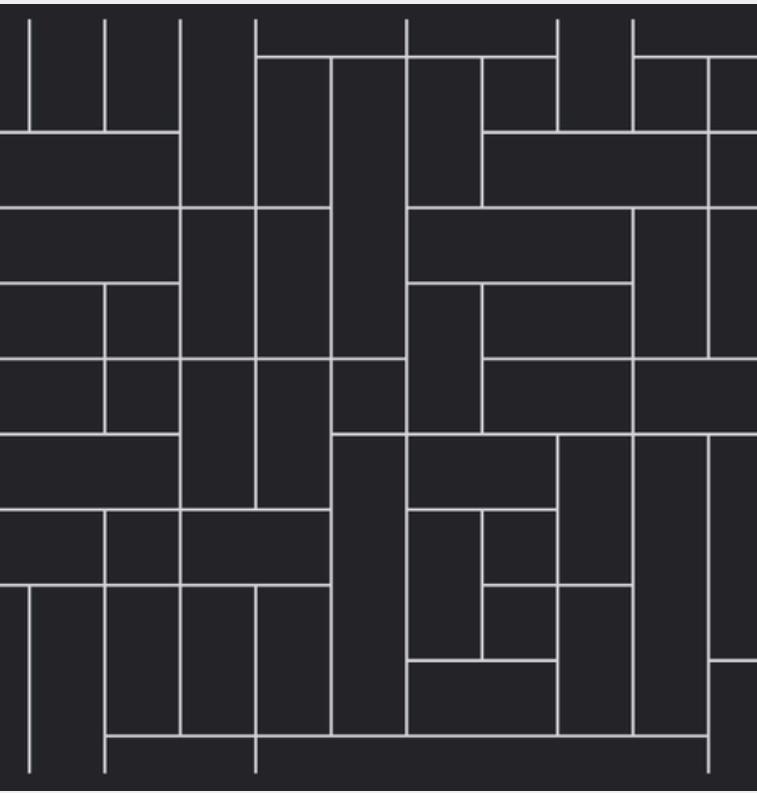
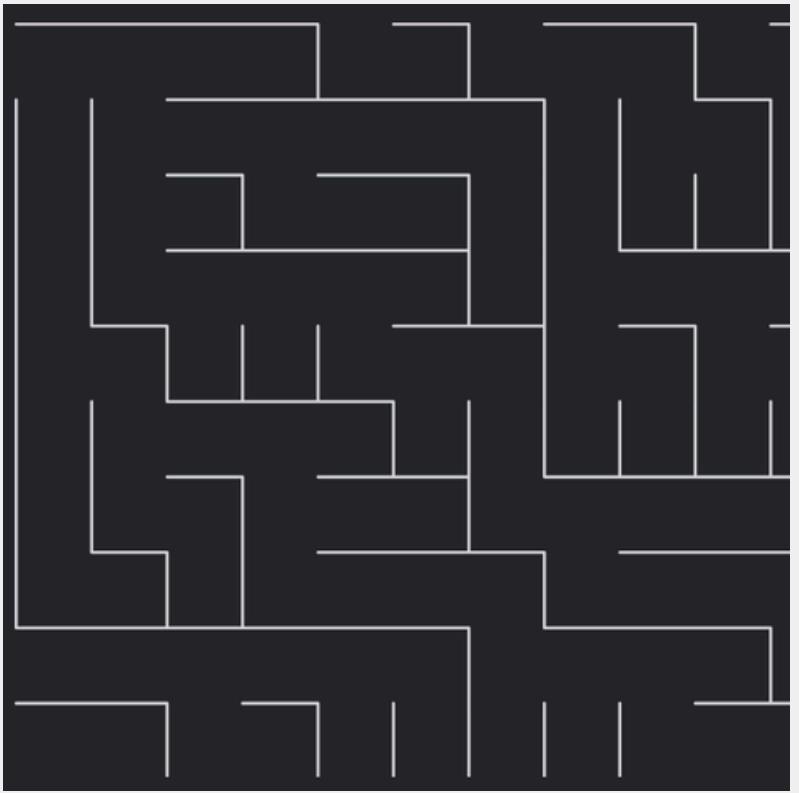
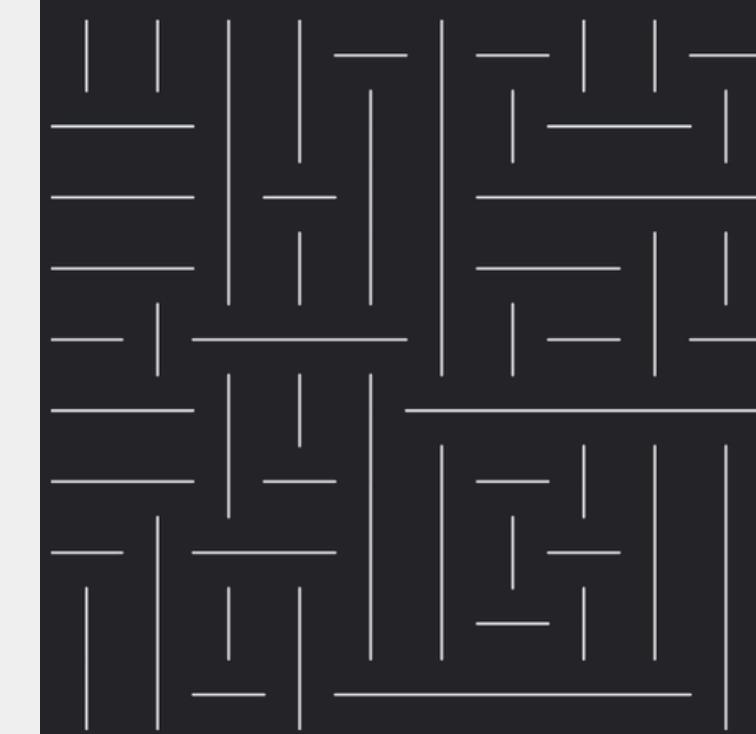
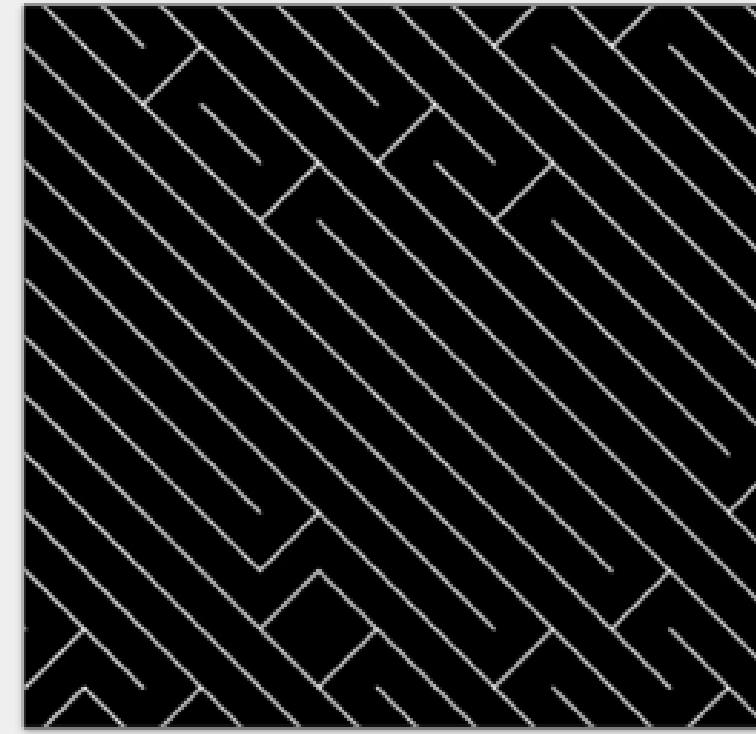
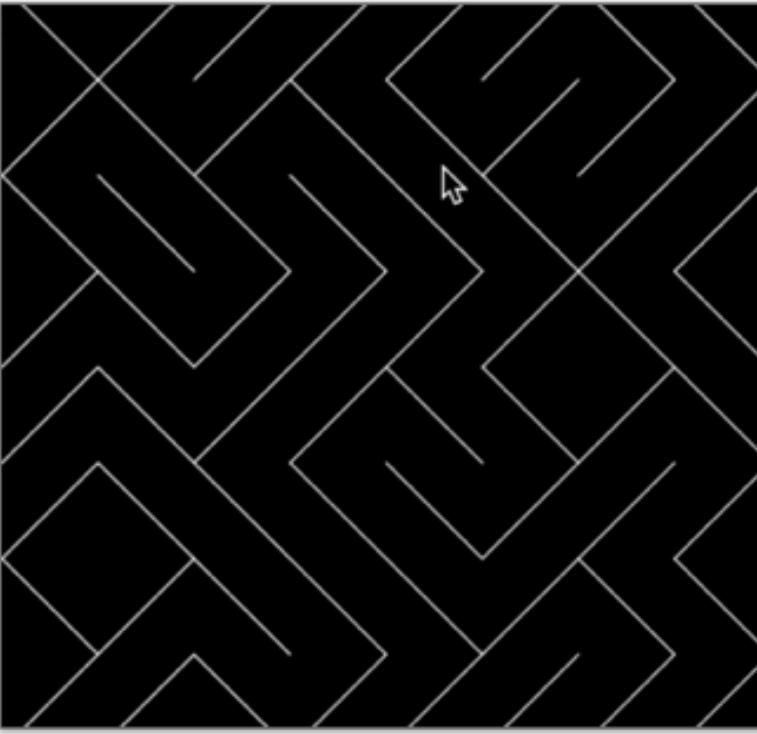
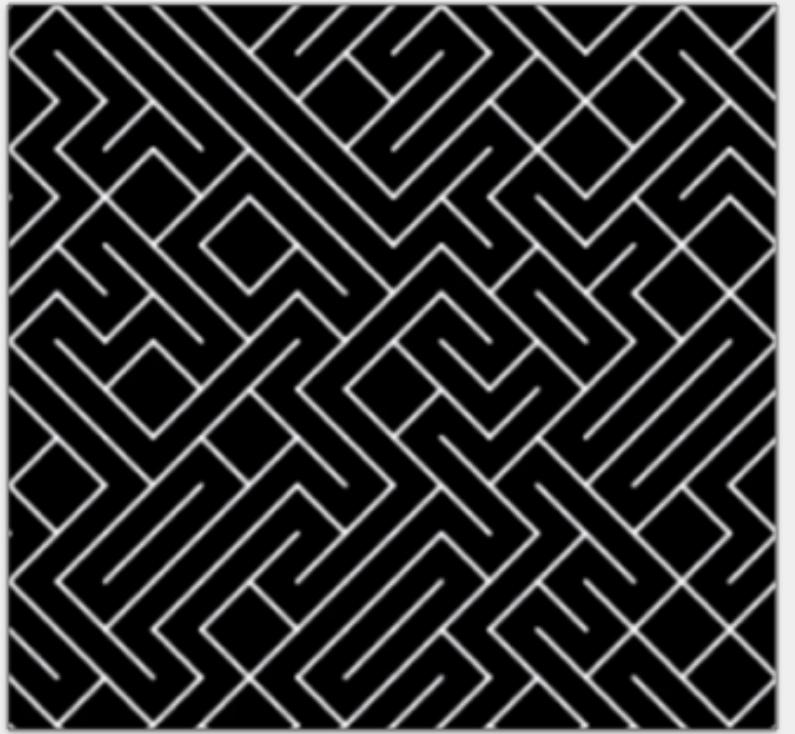
10 Print



Which parameters could we define / change / modulate?

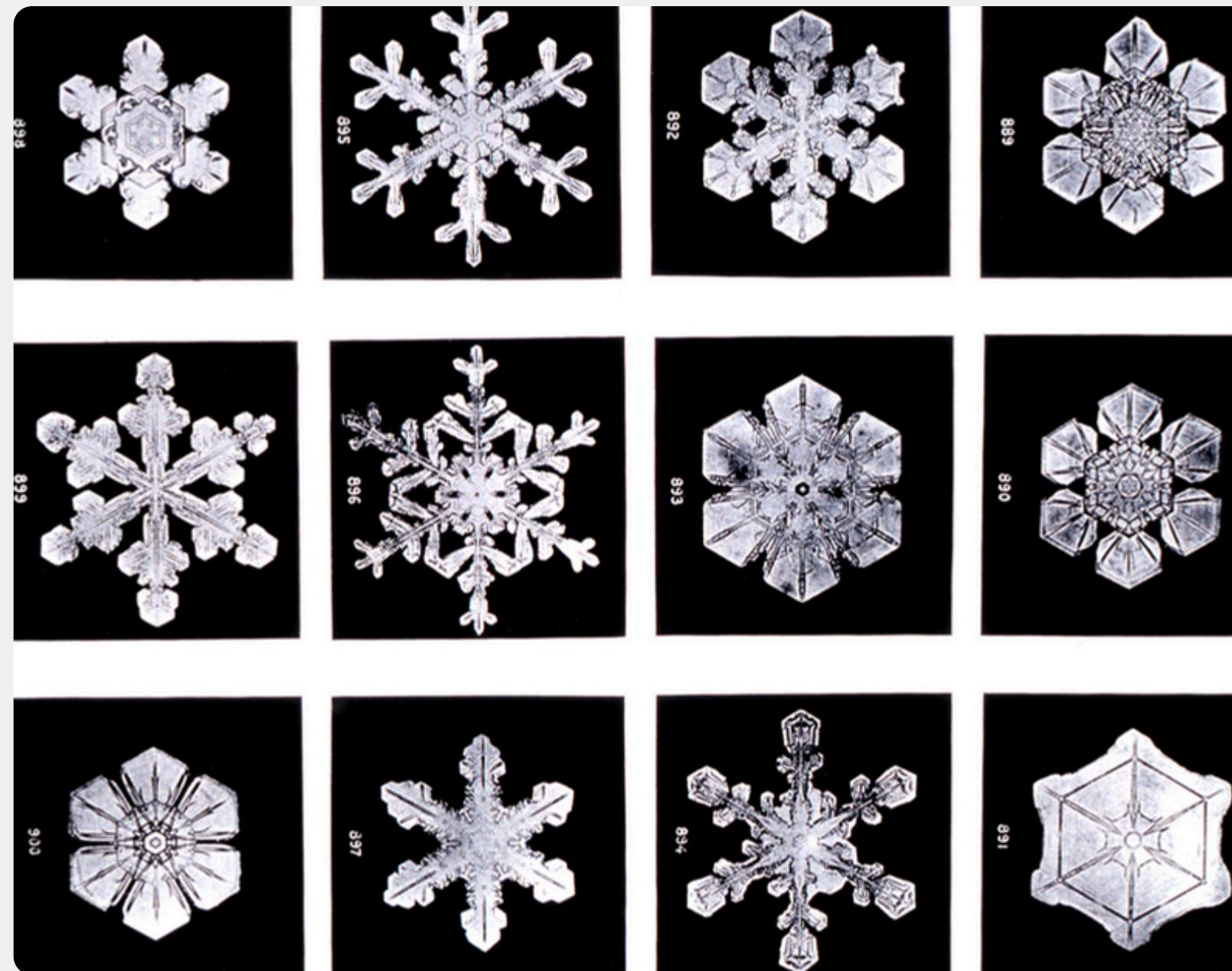
05.

10 Print

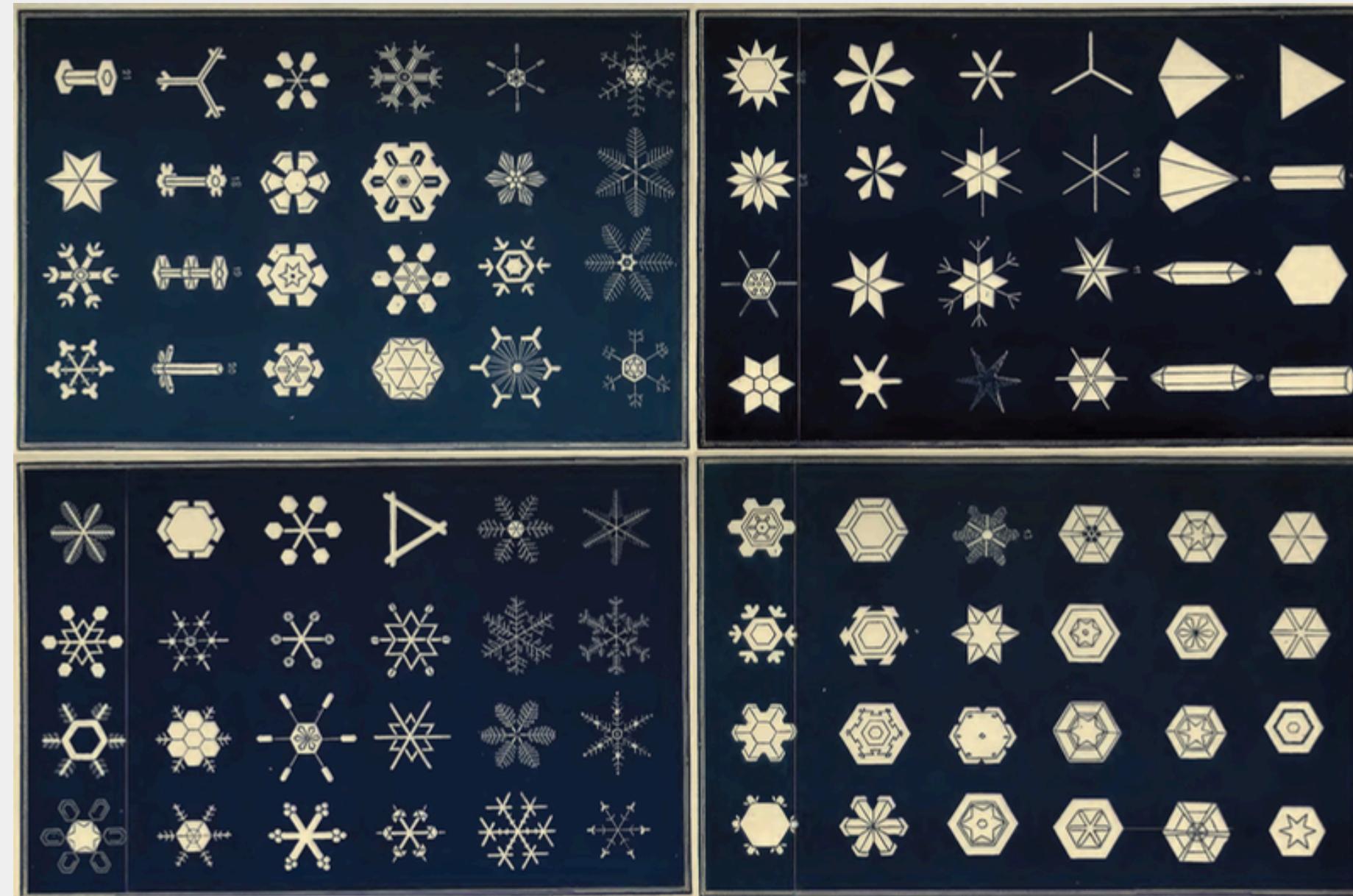


What is Emergence?

Emergence

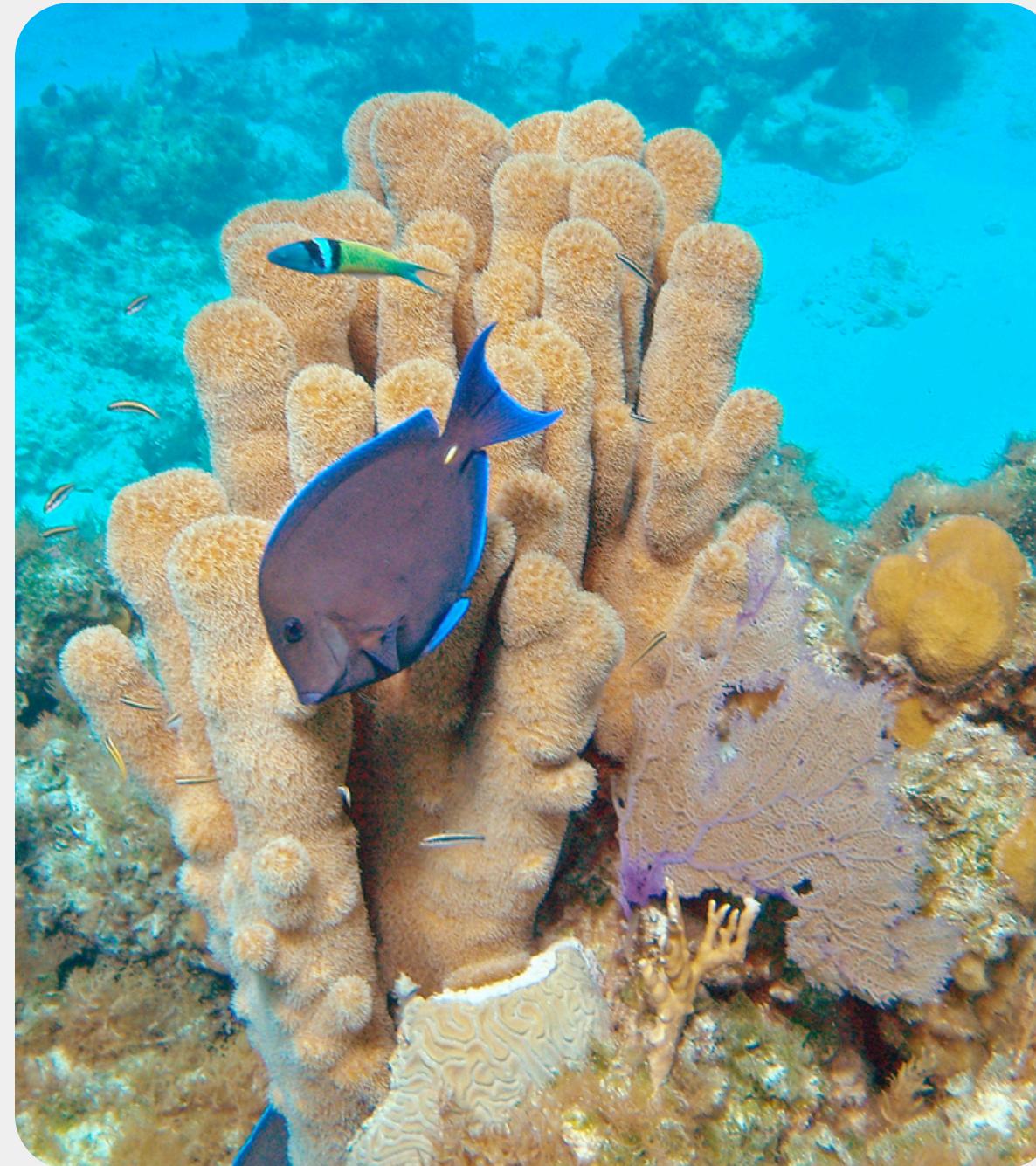


Emergence



An early classification of snowflakes by Israel Perkins Warren.

Emergence



By Mark Peter - Dendrogyra cylindrus (pillar coral) (San Salvador Island, Bahamas) 1, CC BY 2.0, <https://commons.wikimedia.org/w/index.php?curid=39945646>

Emergence



Emergence

In philosophy, systems theory, science, and art, emergence occurs when a complex entity has properties or behaviors that its parts do not have on their own, and emerge only when they interact in a wider whole. [1]

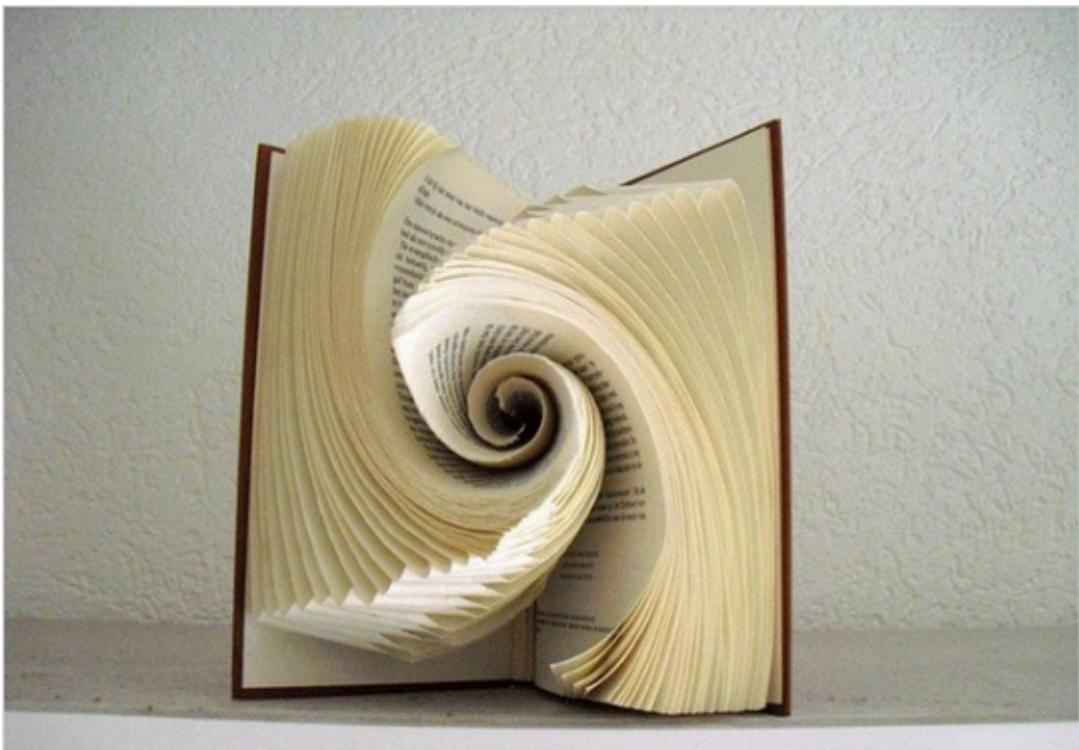
The remarkable simplicity of complexity... [2]

At its core, emergence describes how novel patterns, behaviors, or properties arise from the interactions of simpler parts, such that the whole exhibits something the parts alone don't obviously show. [2]

[1] Wikipedia. 2025. Emergence.

[2] A. Martin and K. Helmerson. 2014. Emergence: the remarkable simplicity of complexity. The Conversation.]

What Does Emergence Mean in the Context of Creative Coding?



A History of Creative Coding

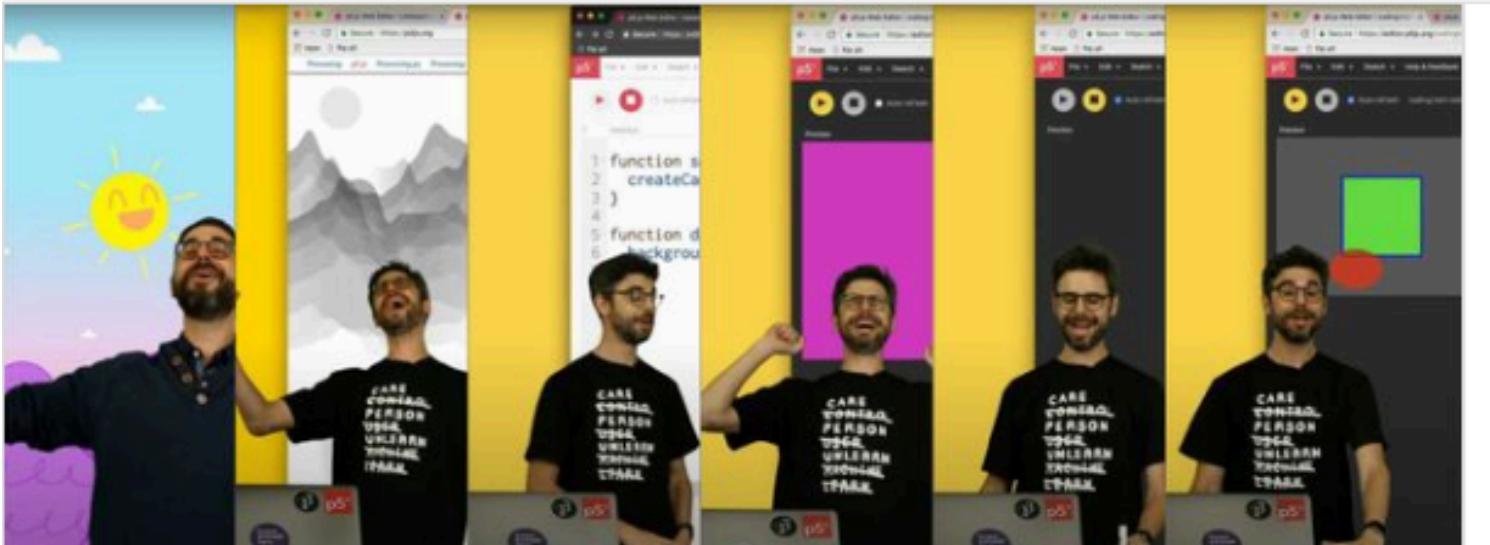
Spoiler — this is not a sprawling timeline article about the history of this field. It's a writeup meant to spur discussion about why there...

Medium / Aug 20, 2018

A Universe in One Line of Code with 10 PRINT

Generating a Universe with One Line of Code

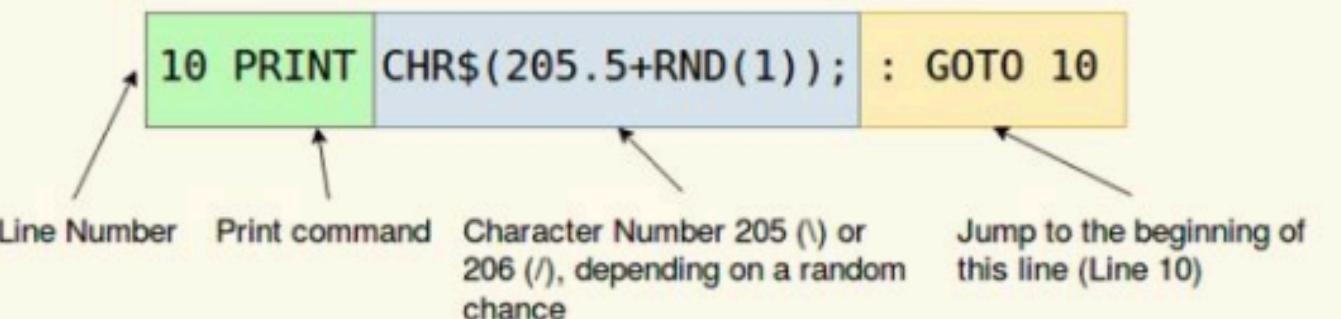
DEV DEV Community



The Coding Train

All aboard the Coding Train with Daniel Shiffman, a YouTube channel dedicated to beginner-friendly creative coding tutorials and challenges.

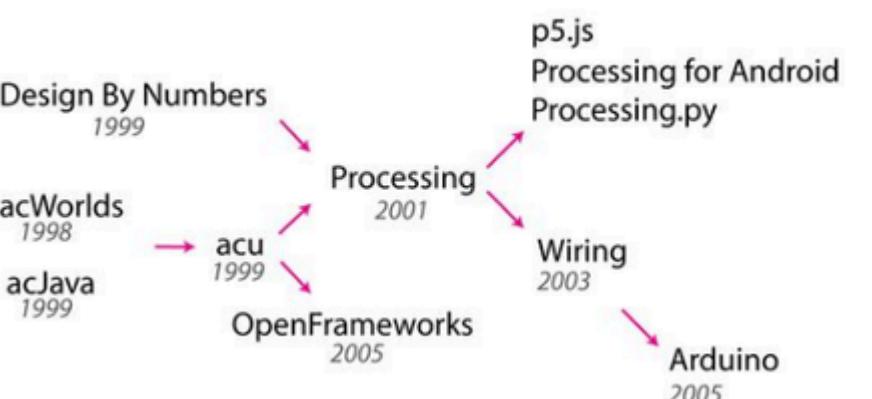
thecodingtrain.com



GOTO 10

About 10 PRINT CHR\$(205.5+RND(1));
: GOTO 10, a book from the MIT Press
10print.org

<https://kosalab.dev/posts/10-print>



History of Processing

Research essay: The History of Processing - Maks Surguy's blog on Technology Innovation, IoT, Design an...

Introduction to generative arts and Processing In this research essay, I will talk about the history and future of a [...]

Maks Surguy's blog on Technology Innovation / Jun 5, 2018



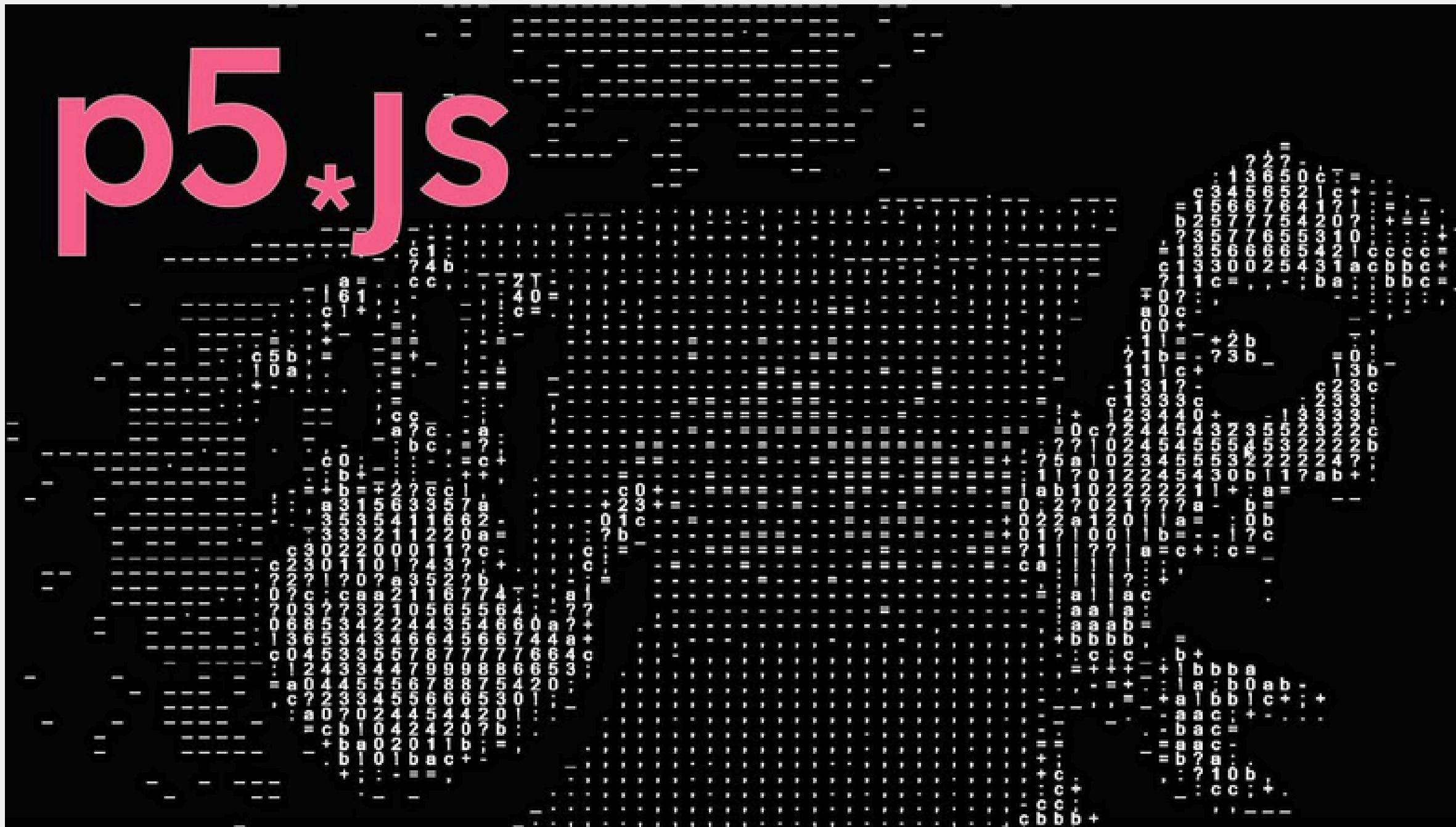
AWESOME CREATIVE CODING

terkelg/awesome-creative-coding: Creative Coding: Generative Art, Data visualization, Interaction Design,...

Creative Coding: Generative Art, Data visualization, Interaction Design, Resources. - terkelg/awesome-creative-coding

[GitHub](#)

Implementation



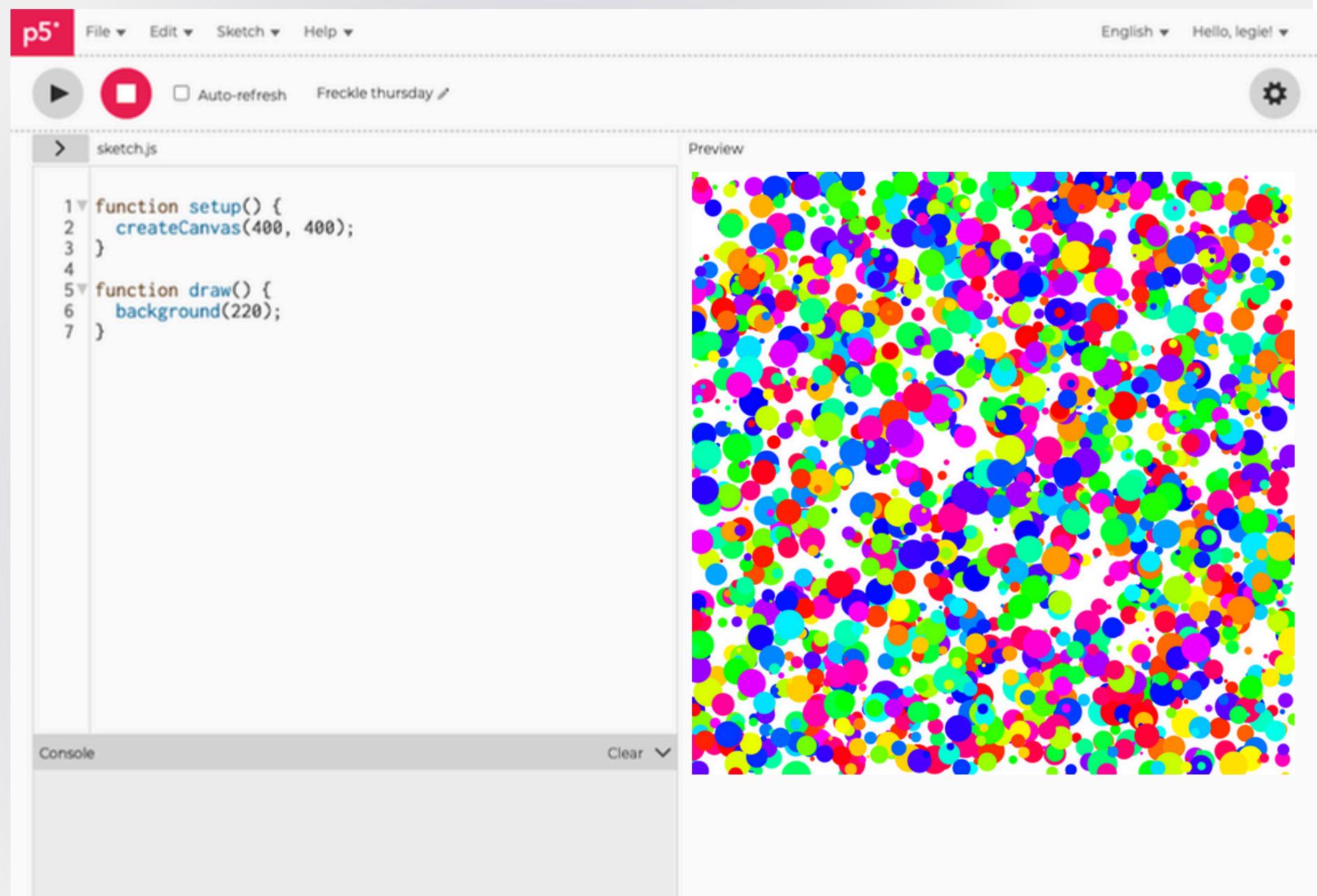
P5.js

The Editor

- p5 comes with a programming environment, the p5.js Editor:
 - easy to set up
 - few external dependencies
 - cross-platform
 - live feedback

The Preview

- A built in code interpreter, which turns our code into visual output



P5.js

p5.js

File Edit Sketch Help English Log in or Sign up

confetti by legie

sketch.js

```
1
2
3 let numCircles = 2000;
4
5 let positionX = [];
6 let positionY = [];
7
8 let stepX = [];
9 let stepY = [];
10
11 let hue = [];
12 let radius = [];
13
14
15
16 function setup() {
17   createCanvas(1000, 1000);
18
19   colorMode(HSB, 360, 100, 100);
20   background(0, 0, 100);
21   fill(200, 100, 100);
22   noStroke();
23
24   // Initialization of the values
25   for (let i = 0; i < numCircles; i++) {
26     positionX[i] = random(0, width);
27     positionY[i] = random(0, height);
28     stepX[i] = random(-5, 5);
29     stepY[i] = random(-5, 5);
30     hue[i] = random(0, 360);
31     radius[i] = random(10, 50);
32   }
33 }
```

Preview

Auto-refresh

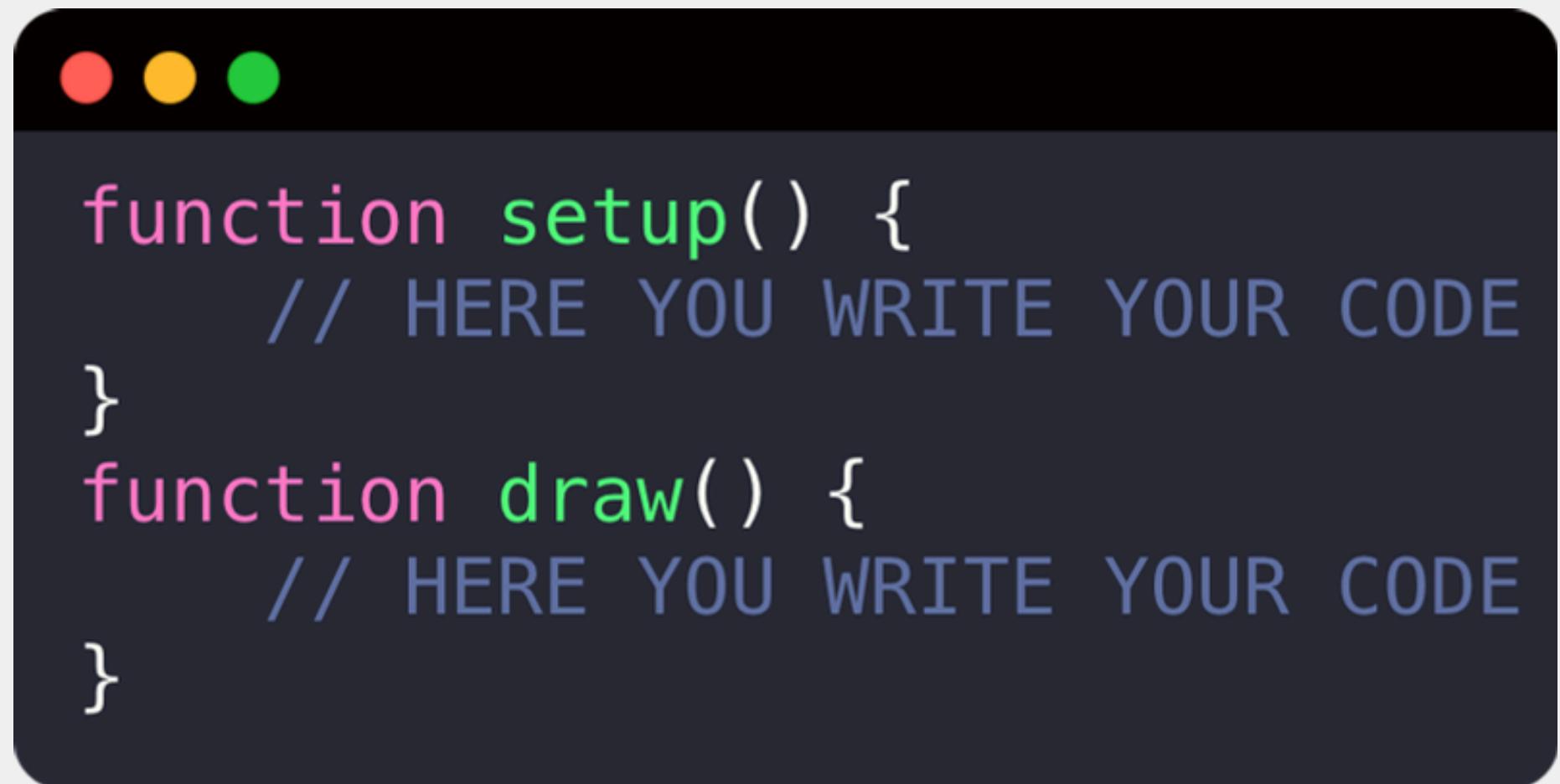
p5.js 1.6.0

<https://editor.p5js.org/legie/full/LyTHREIeS>

P5's Code Structure

All P5 Sketches **must** include at least these two snippets of code, defined by the developers of P5

- **setup()**
 - Executed once when the program is started
- **draw()**
 - Executed as soon as setup() is done
 - Executed again and again until the execution is stopped
 - By default 60 frames in a second



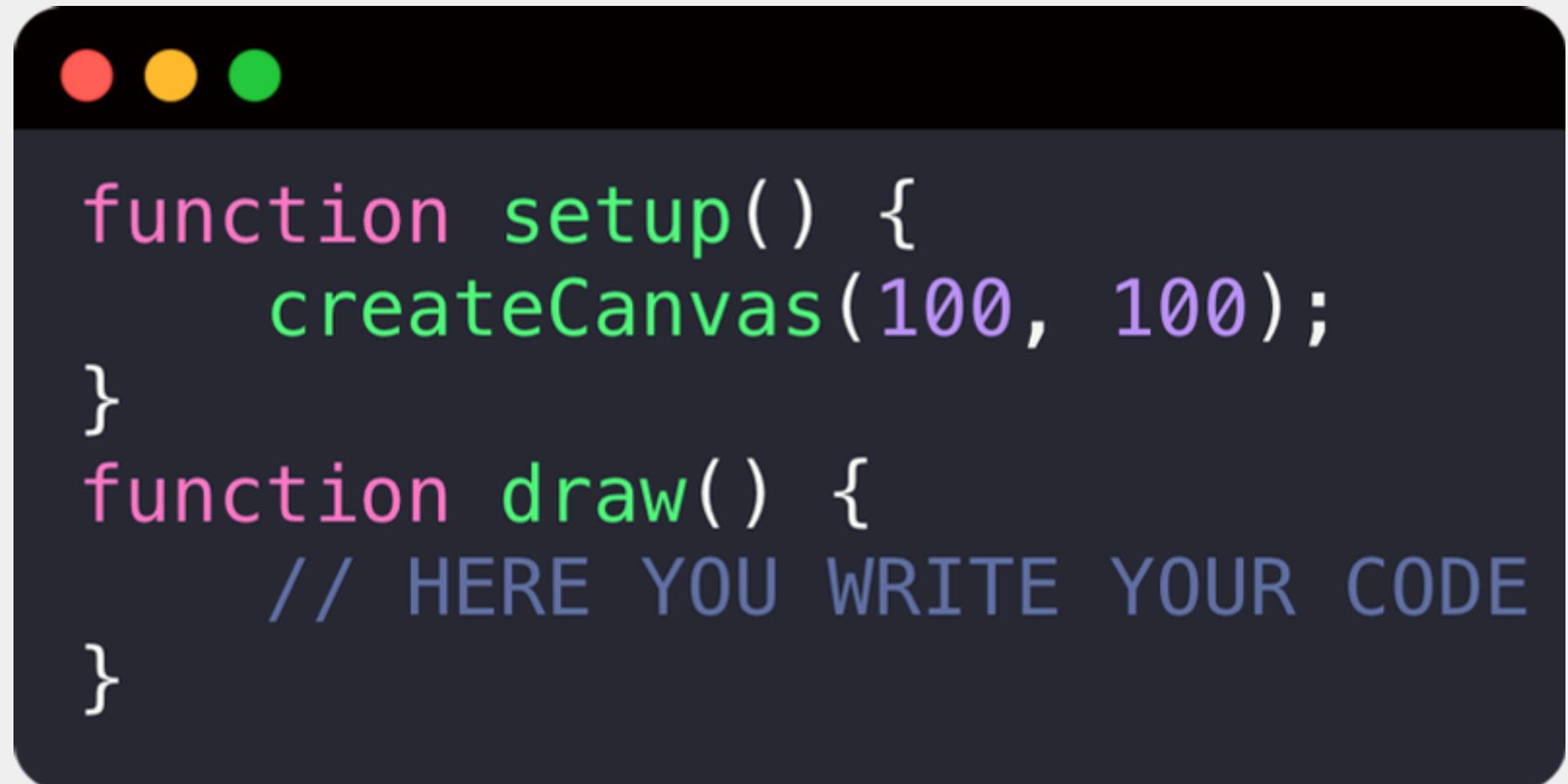
A screenshot of a P5.js code editor window. The window has a dark theme with three colored circular icons in the top-left corner (red, yellow, green). The code area contains the following pseudocode:

```
function setup() {  
    // HERE YOU WRITE YOUR CODE  
}  
function draw() {  
    // HERE YOU WRITE YOUR CODE  
}
```

Drawing - Create Canvas

The **Canvas** is the area on which we will ‘paint’ or ‘draw’ our sketch. It is made of a grid of pixels, which we can assign colors.

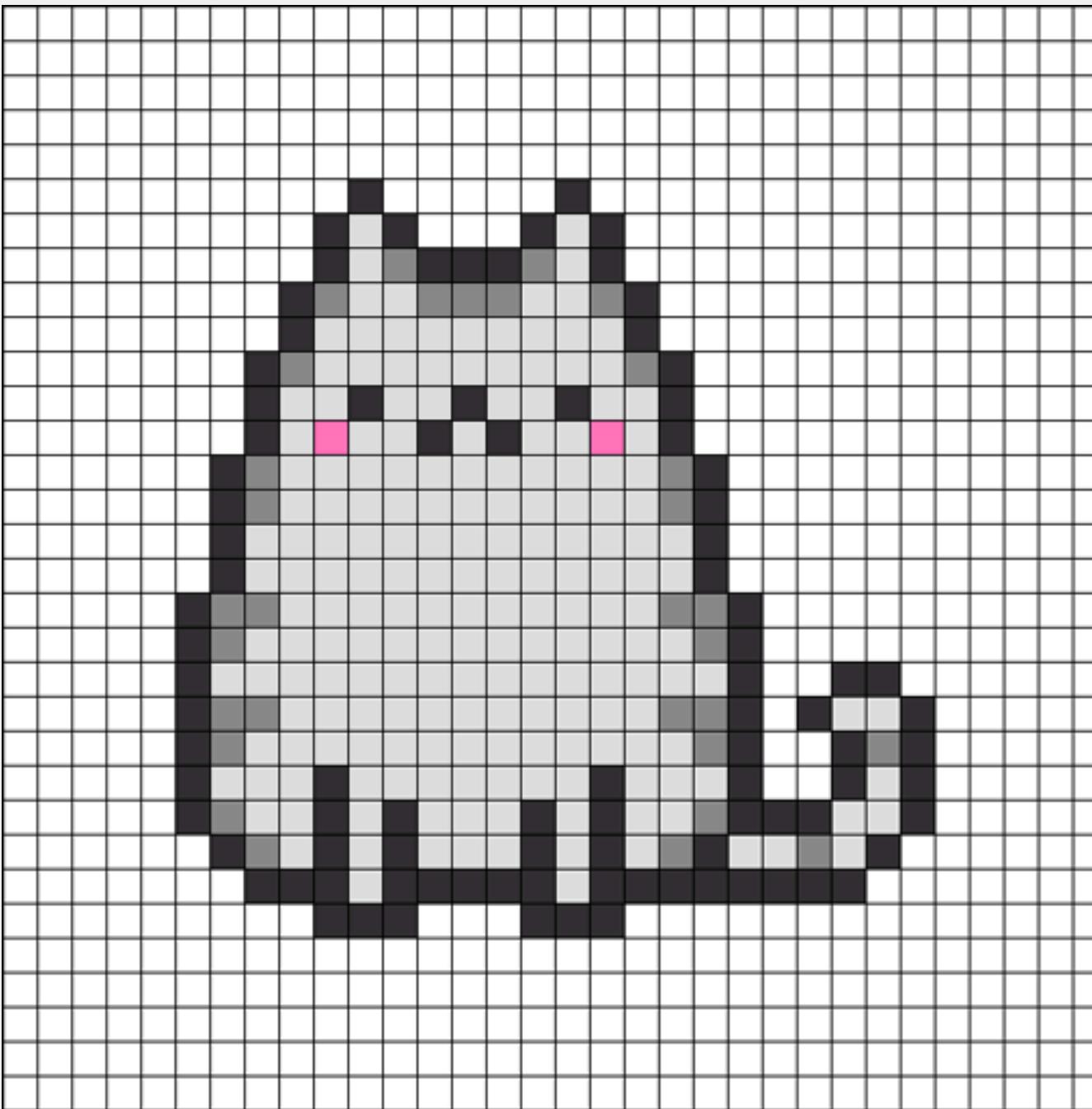
- **createCanvas(width, height)**
 - width and height are in pixels (e.g. 1920px x 1080px).
 - Mostly, we will be working in smaller values, to keep things performant



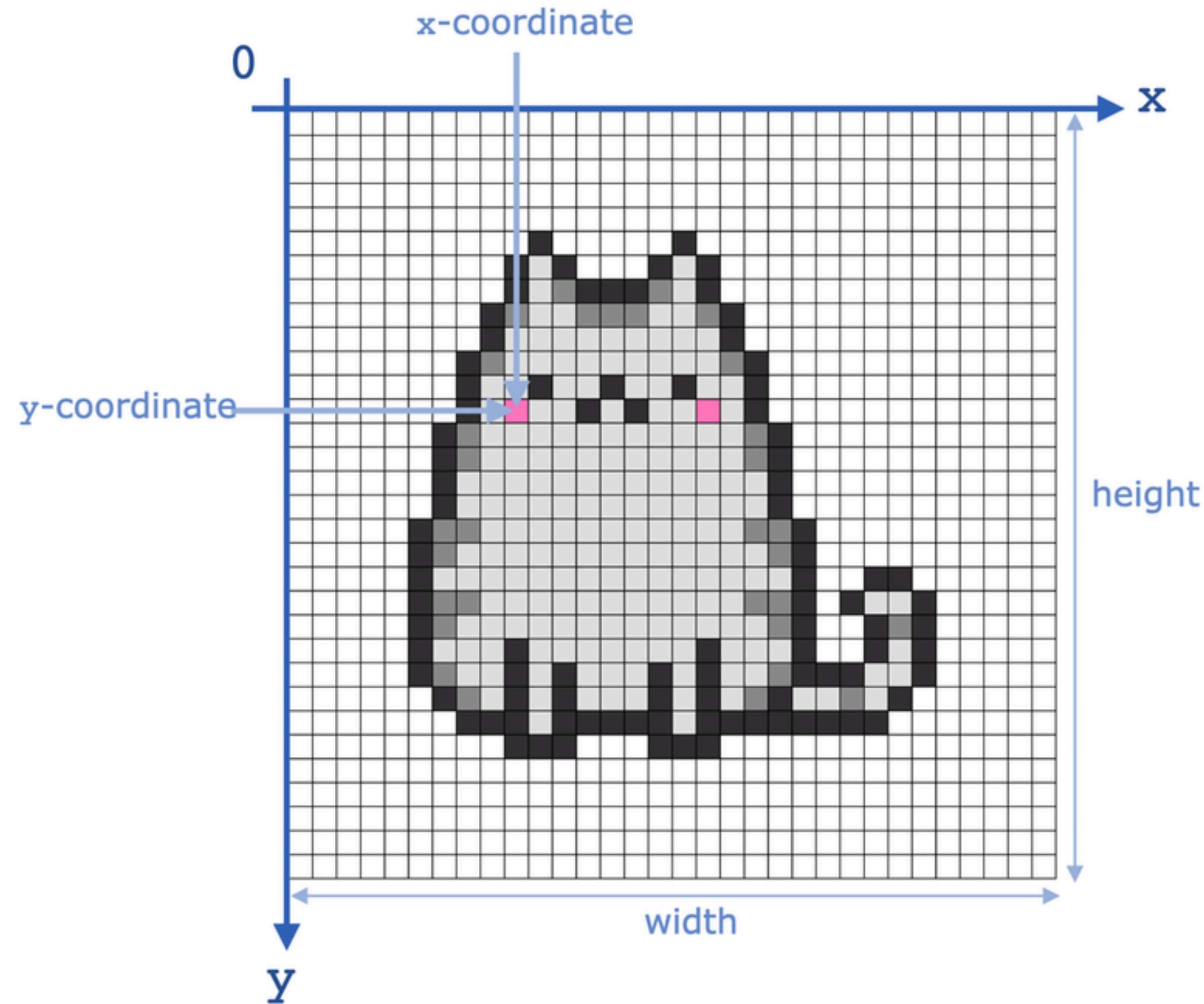
```
function setup() {
  createCanvas(100, 100);
}

function draw() {
  // HERE YOU WRITE YOUR CODE
}
```

Drawing - Create Canvas



Drawing - Create Canvas

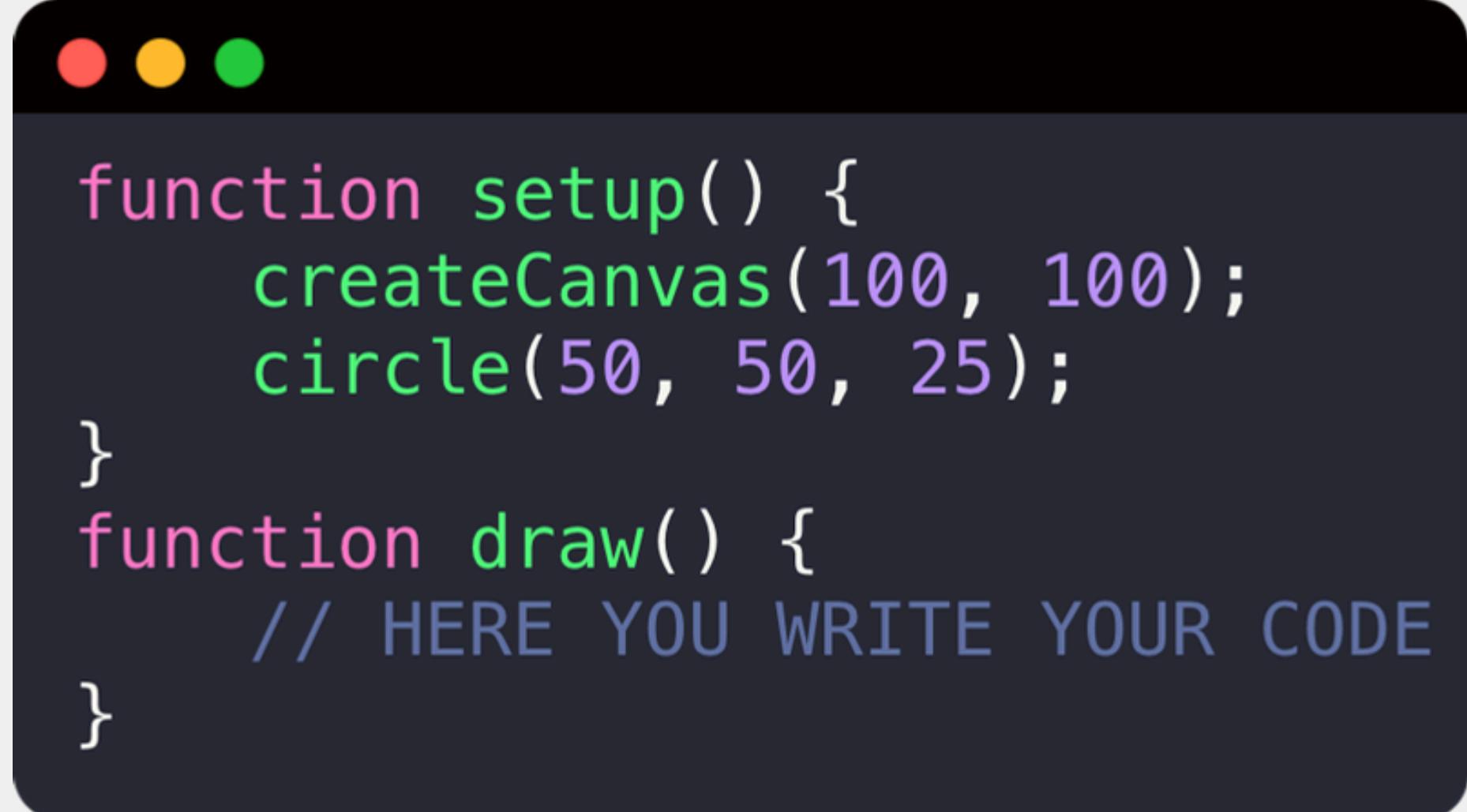


Calling Functions

Functions are small snippets of code that encapsulate a set of instructions. Generally, they are modular units of code designed to perform specific tasks

A typical drawing function call could look for example as follows:

- **circle(center x, center y, diameter)**
 - The order of the parameters is fixed and must be followed!
 - Again, values in px



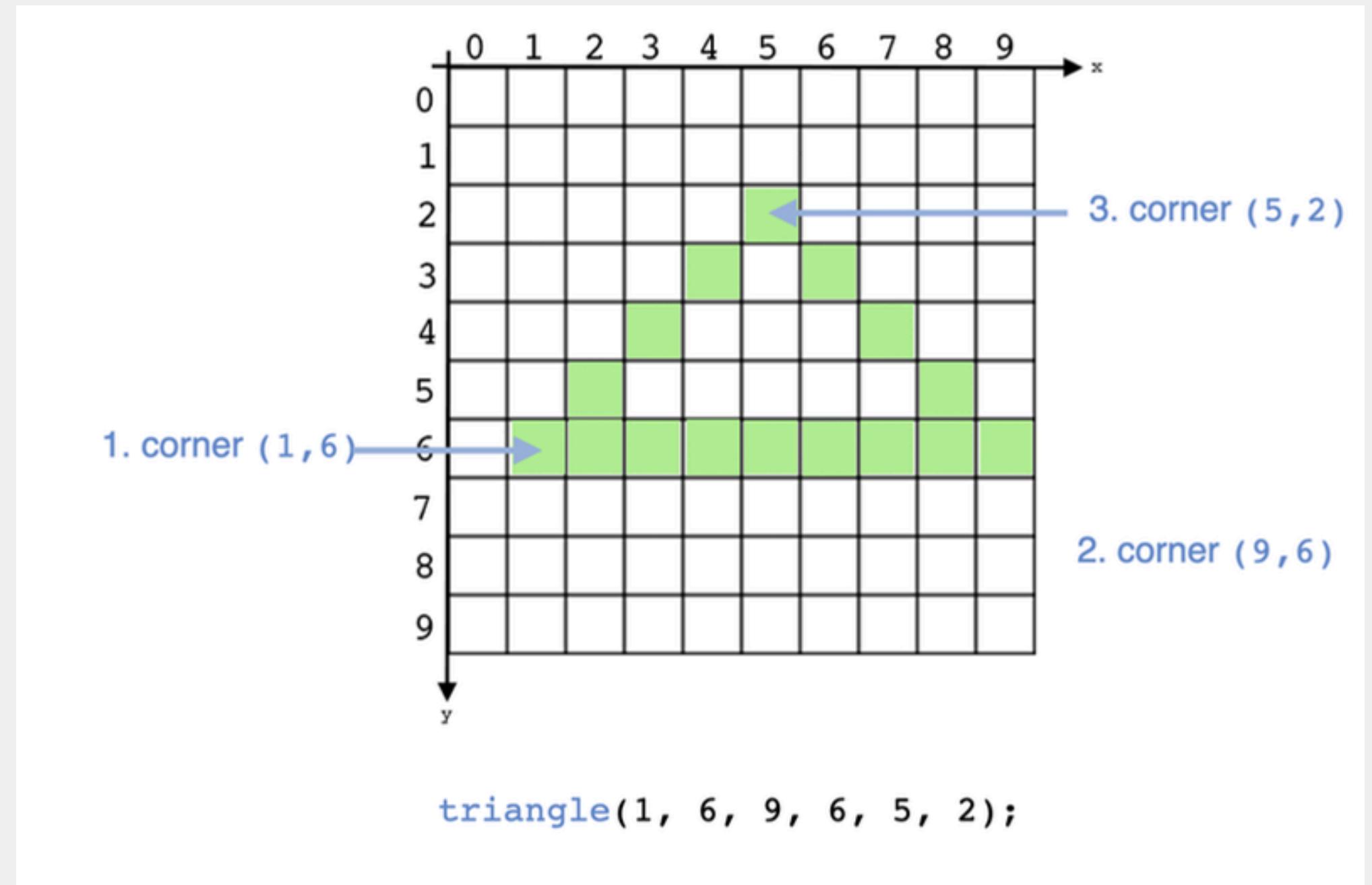
```
function setup() {
  createCanvas(100, 100);
  circle(50, 50, 25);
}

function draw() {
  // HERE YOU WRITE YOUR CODE
}
```

Calling Functions

Triangle

- `triangle(x1, y1, x2, y2, x3, y3);`
 - Arguments:
 - corner (x1, y1)
 - corner (x2, y2)
 - corner (x3, y3)

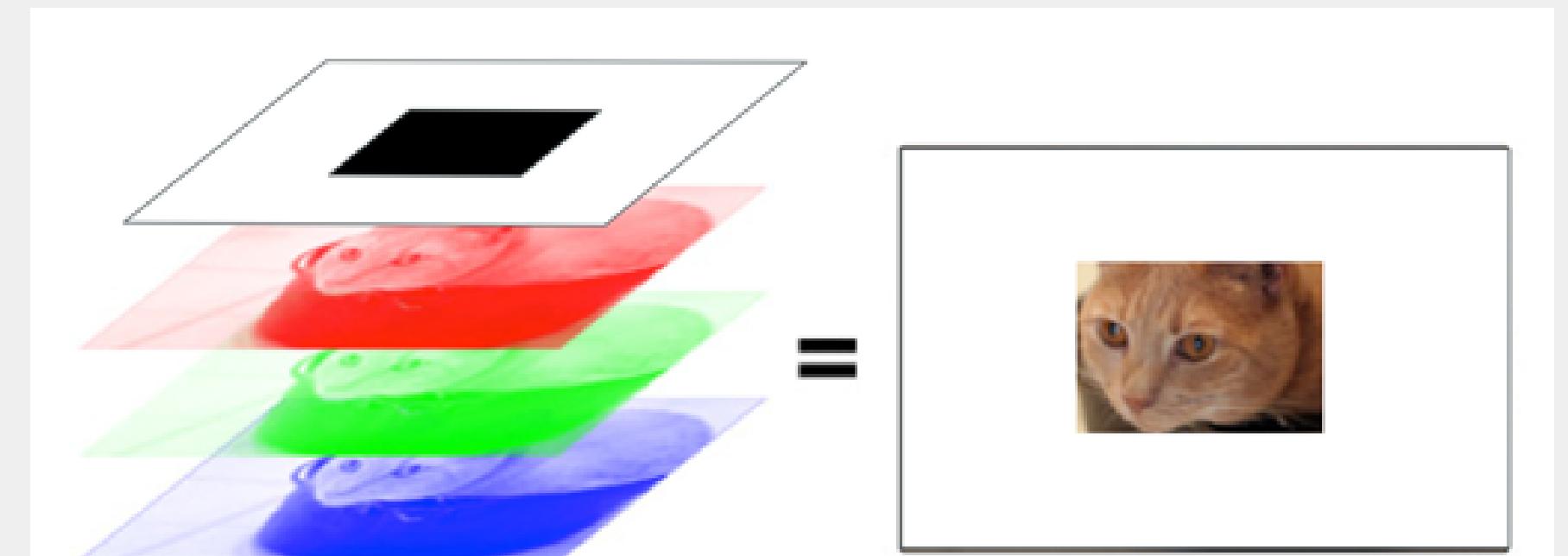
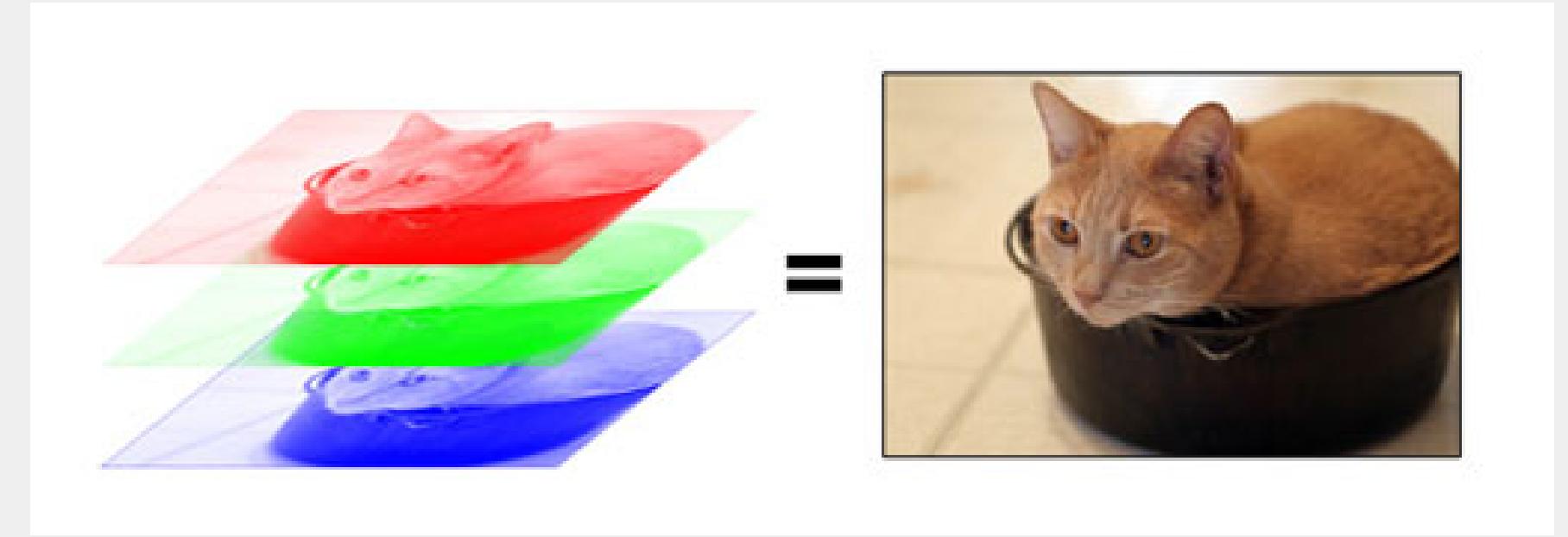


2D Primitives

- **arc()**
 - **ellipse()**
 - **circle()**
 - **line()**
 - **point()**
 - **quad()**
 - **rect()**
 - **square()**
 - **triangle()**
-
- **Polygons**
 - **Images, Videos**
 - **etc.**

Colors

- RGBA and HSB
- The default
 - red, green, blue, alpha
 - 0 ... 255
 - 0 = no color, 255 = full saturation
 - Alpha is optional *
 - 0 = fully transparent,
 - 255 = fully opaque



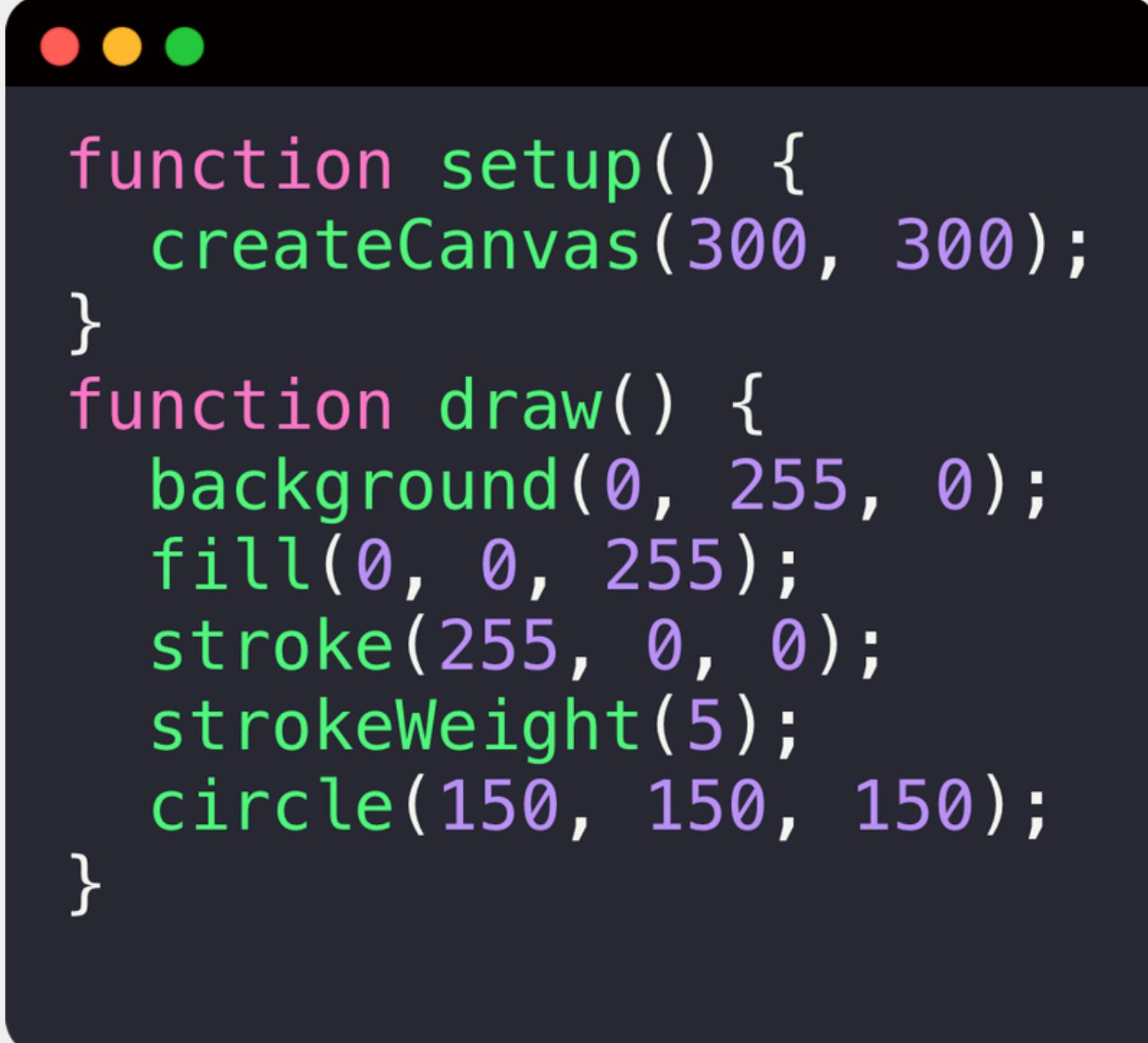
Color Function Calls

- Setting the background color:

```
background(r, g, b);
```

- Changing attributes of the drawing commands:

```
fill(r, g, b);
stroke(r, g, b);
strokeWeight(w);
noFill();
noStroke();
```

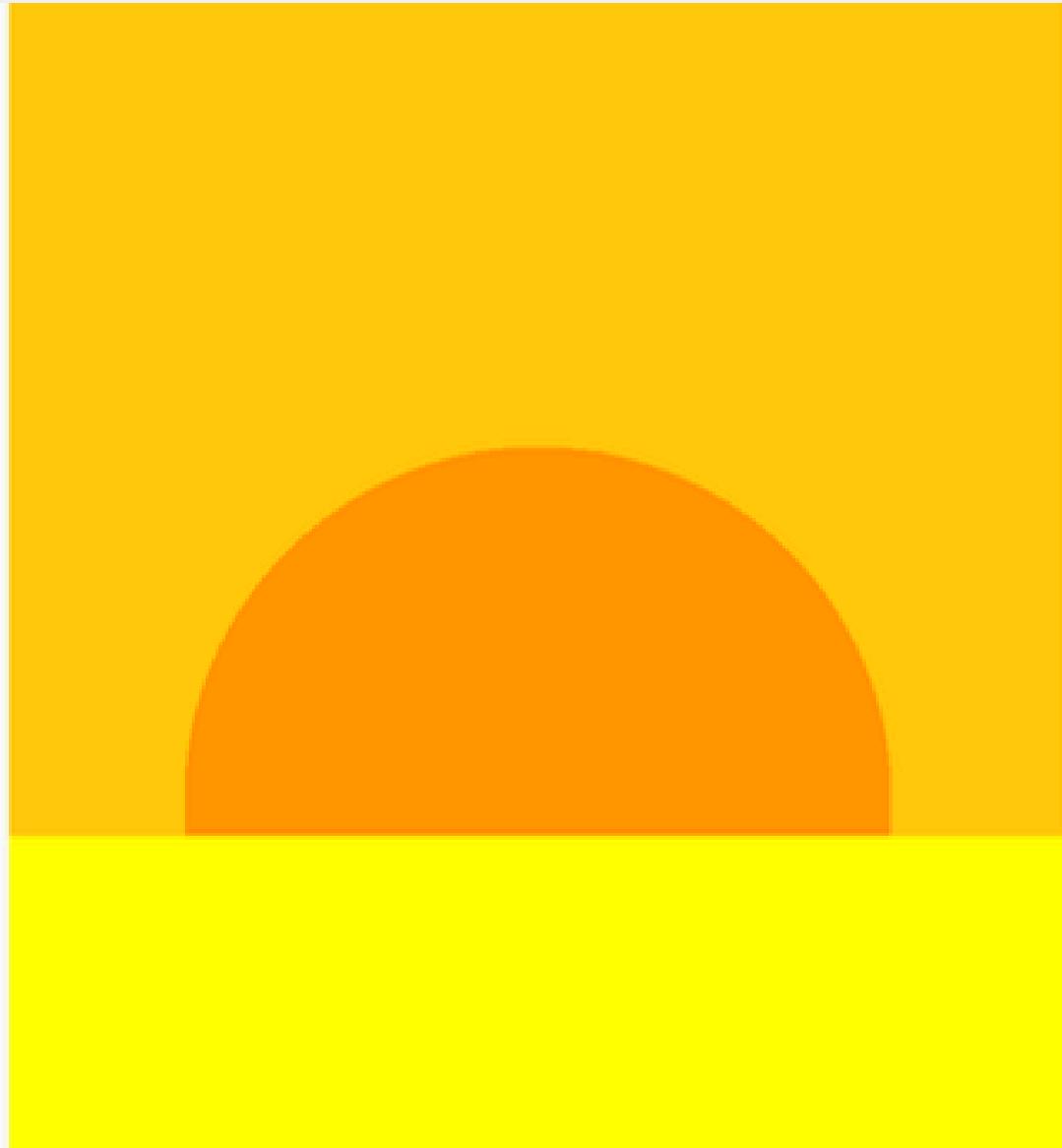


```
function setup() {
  createCanvas(300, 300);
}

function draw() {
  background(0, 255, 0);
  fill(0, 0, 255);
  stroke(255, 0, 0);
  strokeWeight(5);
  circle(150, 150, 150);
}
```

Example

```
1  function setup() {  
2      createCanvas(300, 400);  
3      // Background color of the canvas  
4      background(255, 200, 10);  
5  }  
6  
7  function draw() {  
8      // Ellipse  
9      fill(255, 150, 0);  
10     noStroke();  
11     ellipse(150, 240, 200, 200);  
12  
13     // Rectangle ground  
14     fill(255, 255, 0);  
15     rect(0, 250, 500, 250);  
16 }
```



<https://editor.p5js.org/legie/sketches/1evl31ovv>
https://editor.p5js.org/legie/sketches/oLR_BdLGu

How can we remember all this?

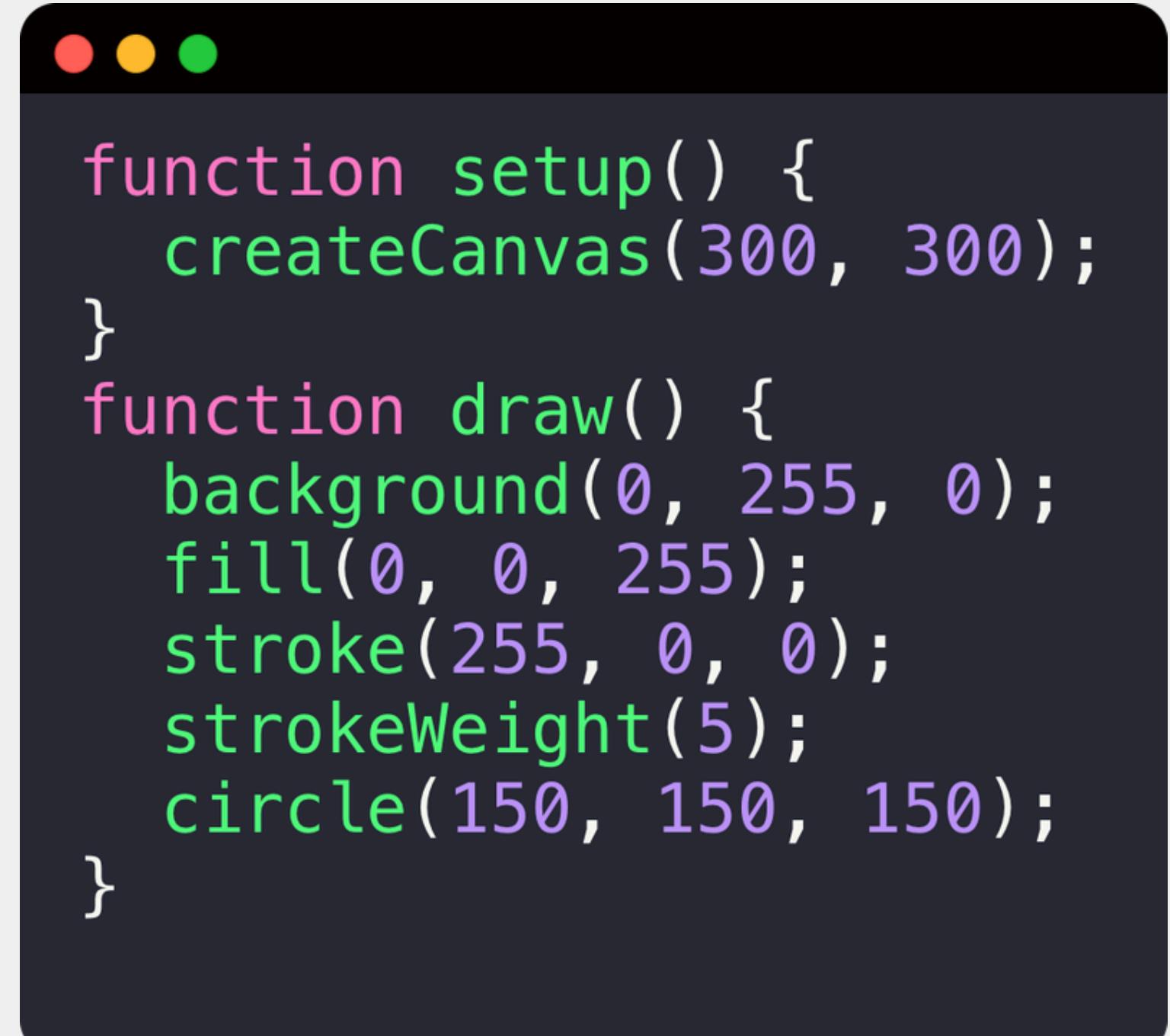
Color Function Calls

- Setting the background color:

```
background(r, g, b);
```

- Changing attributes of the drawing commands:

```
fill(r, g, b);
stroke(r, g, b);
strokeWeight(w);
noFill();
noStroke();
```



```
function setup() {
  createCanvas(300, 300);
}

function draw() {
  background(0, 255, 0);
  fill(0, 0, 255);
  stroke(255, 0, 0);
  strokeWeight(5);
  circle(150, 150, 150);
}
```

Read the documentation :)

The documentation is your friend

Look it up in the reference: [p5.js documentation](#)

- The reference is like a dictionary of the programming language.
- Learning to work with the reference is as important as learning the programming syntax.
- I recommend to bookmark this page as you will come back to it many, many times.

Implementing 10 PRINT

- What do we need to implement a visual pattern such as the 10 PRINT pattern in p5.js?

Algorithm:

- Go row by row
- Place a / or a \...
- ...randomly
-



Loops

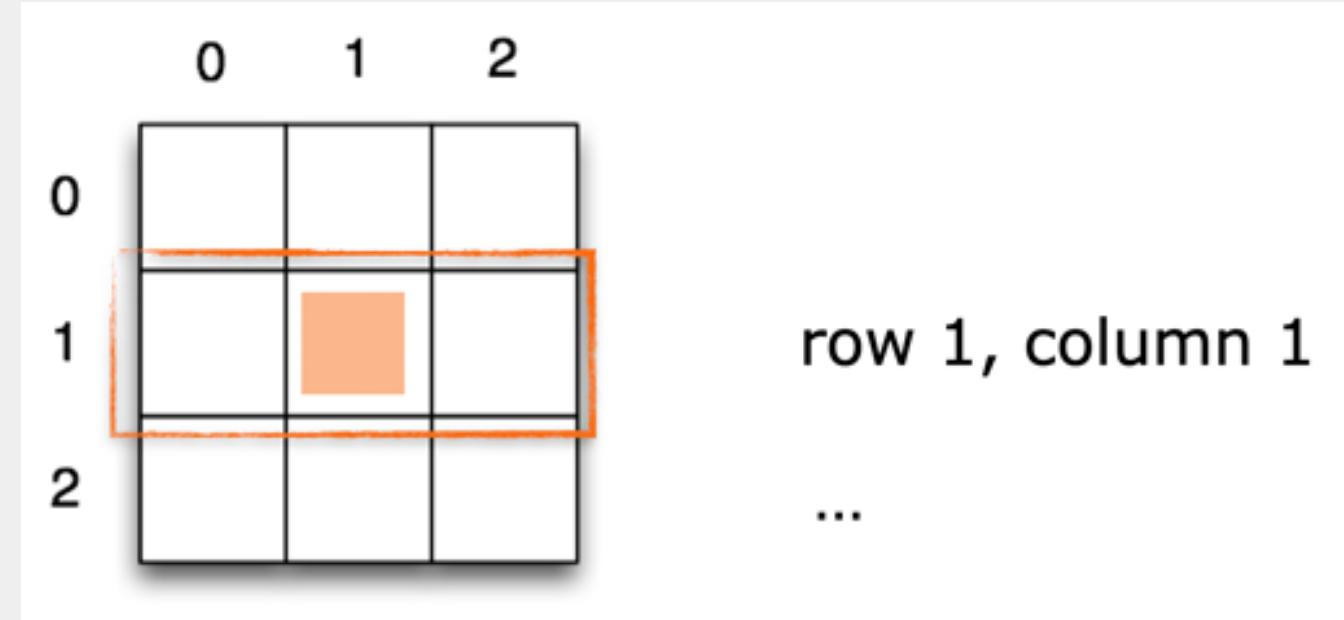
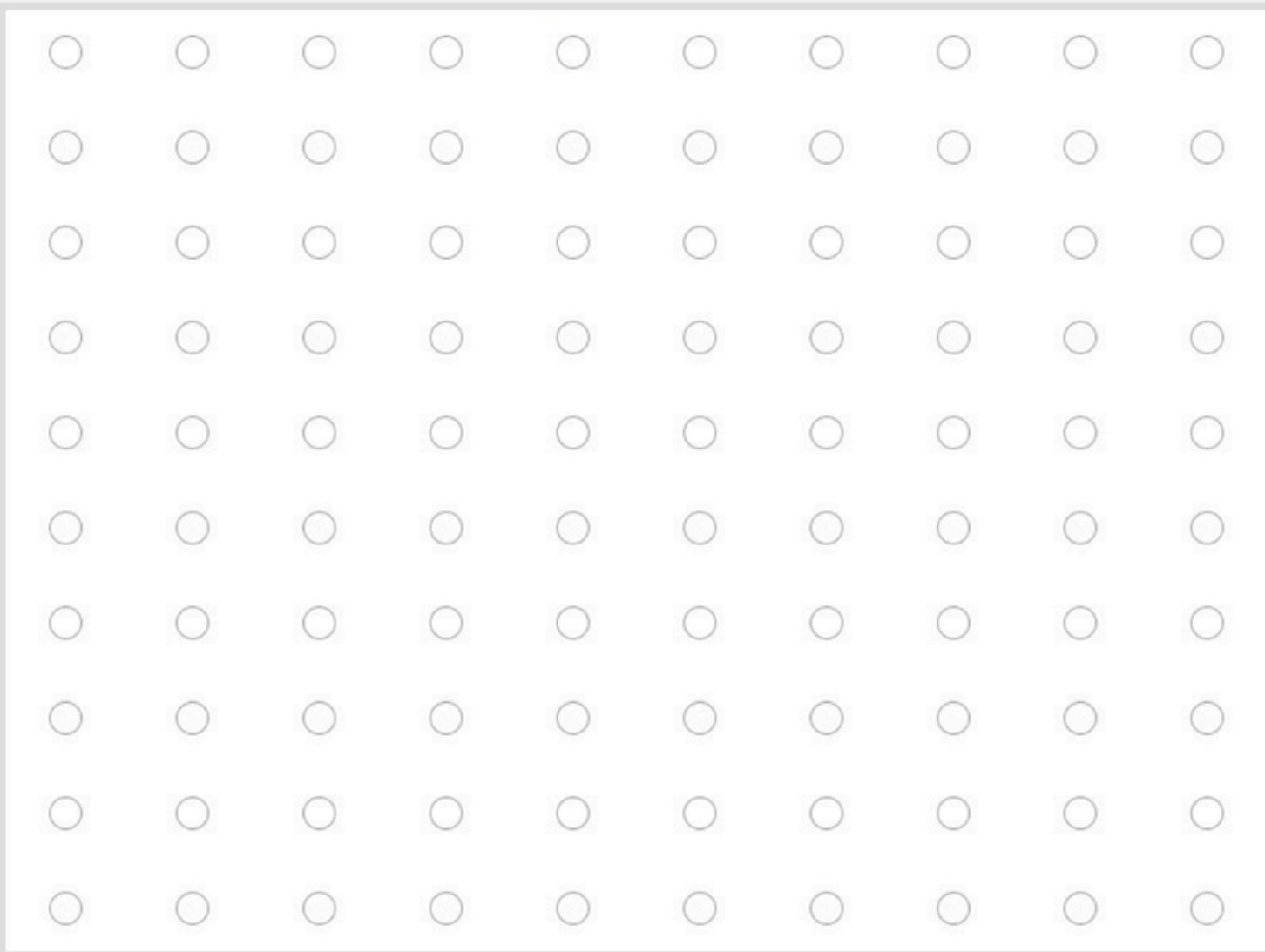
- **Iterators:**
 - any variable, i is typical
- **End condition**
 - you can chose any end value, or variable
- **Step size**
 - $i++ = i + 1$
- **Scope**
 - variables only exist within the loop function

```
● ● ●  
for (let i = 0; i < number0fTimes; i++) {  
    // code  
}
```

```
● ● ●  
for (let i = 0; i < 3; i++) {  
    print("wow");  
}
```

Example

Loops



// Pseudo Code
For every row {
 For every column {
 }
}

The code block shows a dark-themed window with three circular window controls at the top. It contains pseudo code for nested loops, starting with a double slash // followed by the text.

Loops

- **Iterators:**
 - any variable, i is typical
- **End condition**
 - you can chose any end value, or variable
- **Step size**
 - $i++ = i + 1$
- **Scope**
 - variables only exist within the loop function

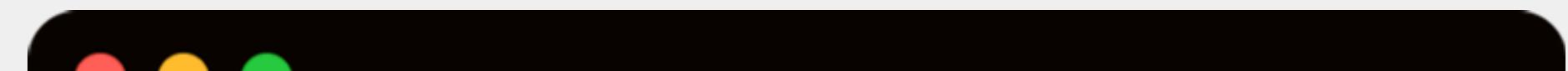
```
● ● ●  
for (let i = 0; i < number0fTimes; i++) {  
    // code  
}
```

```
● ● ●  
for (let i = 0; i < 3; i++) {  
    print("wow");  
}
```

Example

Implementing 10 PRINT

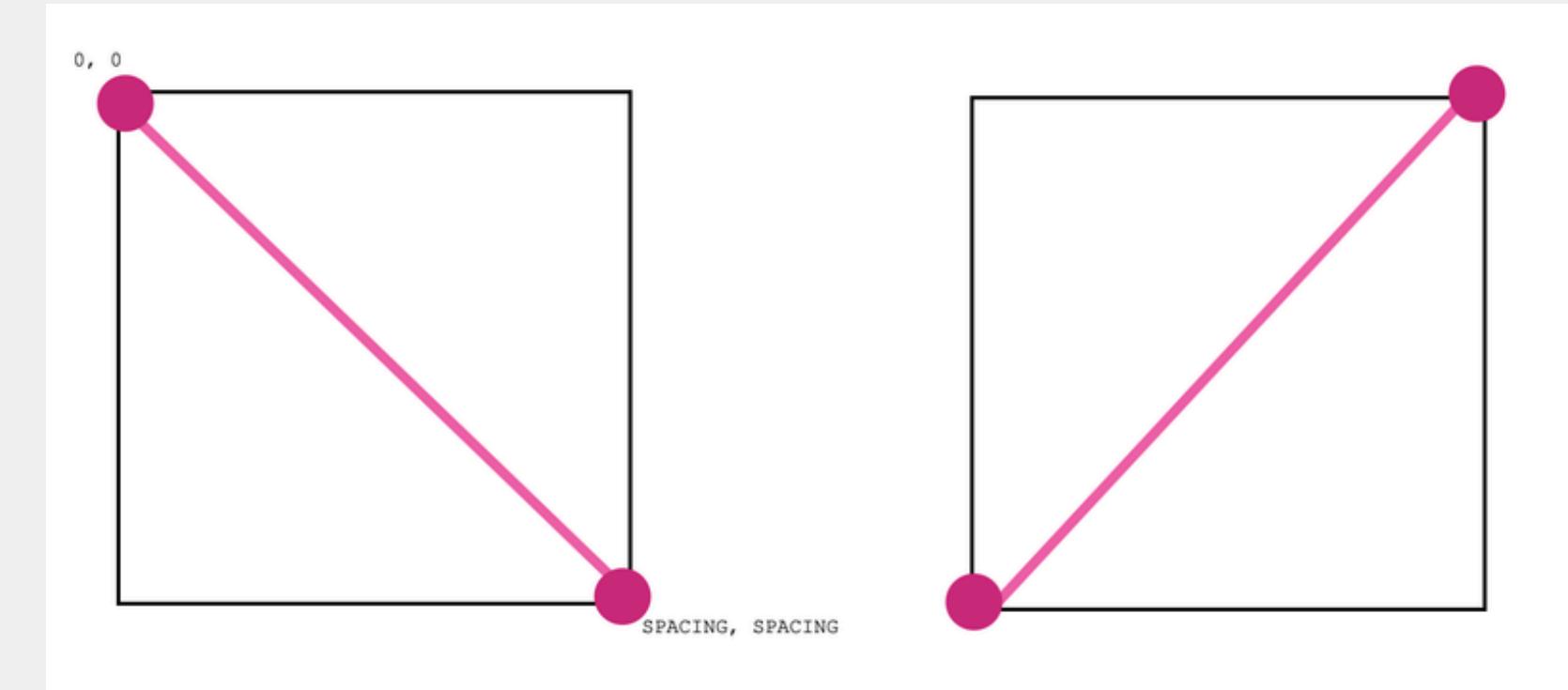
- What do we need to implement a visual pattern such as the 10 PRINT pattern in p5.js?



```
// https://editor.p5js.org/legie/sketches/vyn6r7pxG
line(x, y, x + SPACING, y + SPACING);
// OR
line(x, y + SPACING, x + SPACING, y);
```

Algorithm:

- Go row by row ✓
- Place a / or a \...
- ...randomly



Implementing 10 PRINT

- What do we need to implement a visual pattern such as the 10 PRINT pattern in p5.js?

Algorithm:

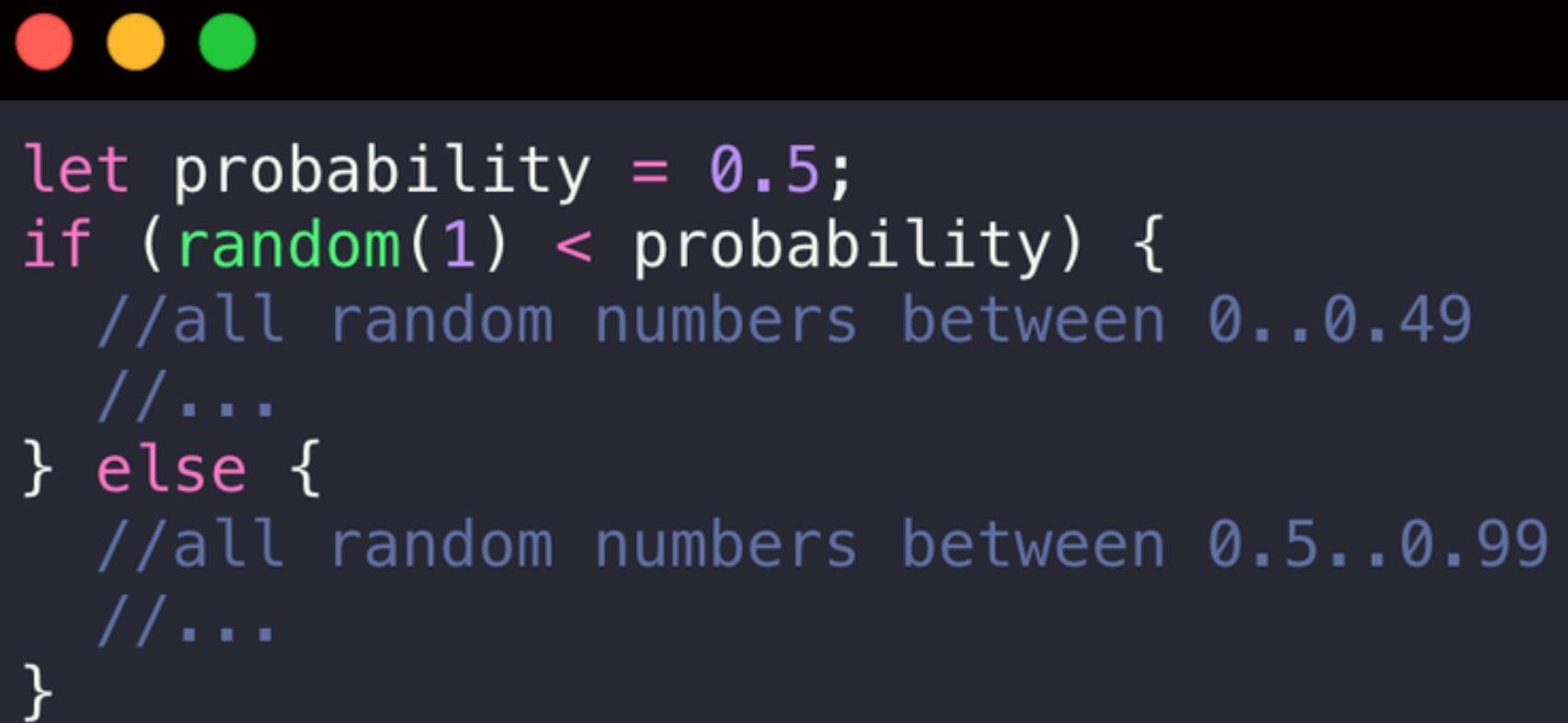
- Go row by row ✓
- Place a / or a \... ✓
- ...randomly ??
-



Randomness

- random(1); → gives random numbers between 0..0.99

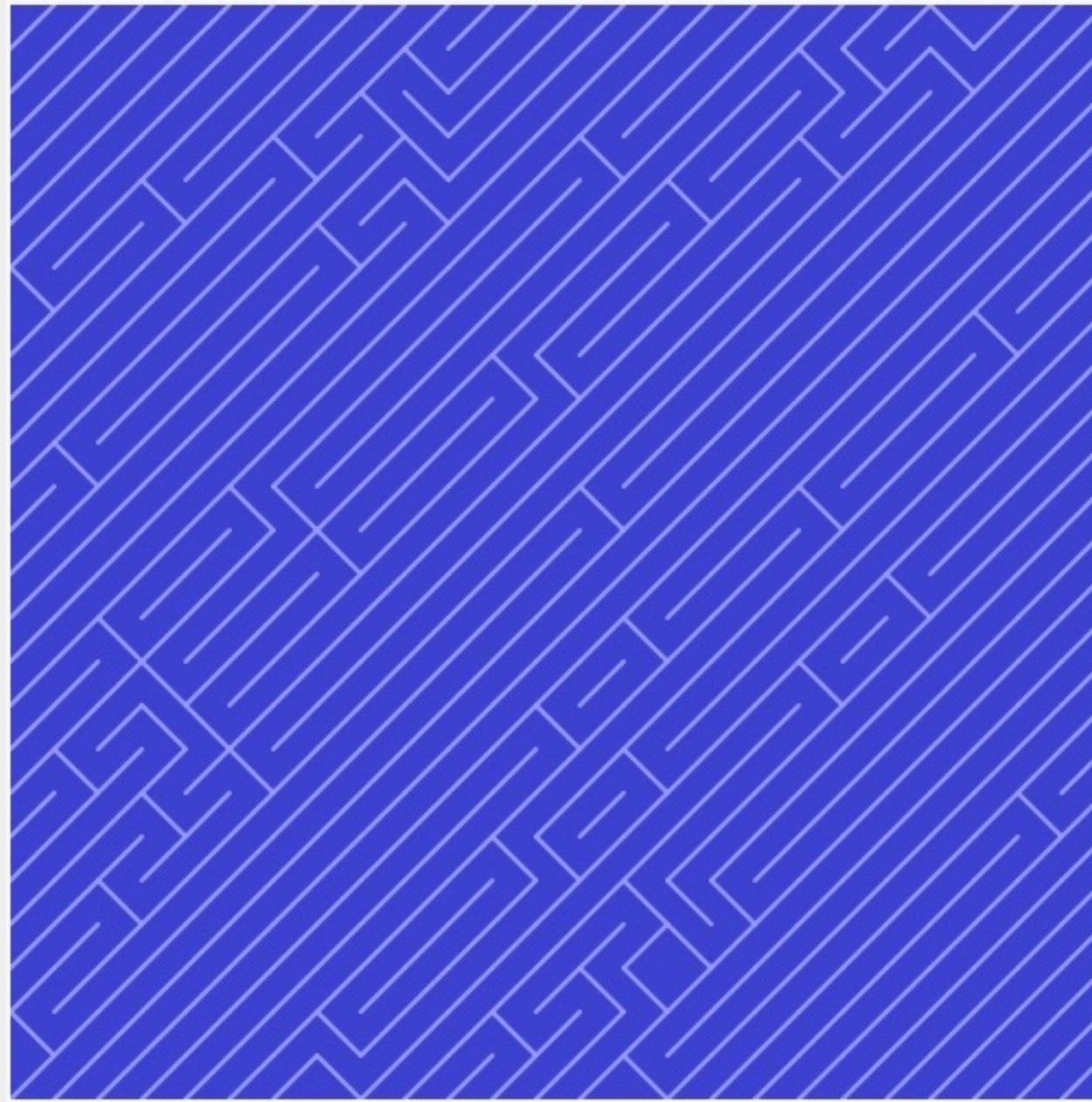
```
let probability = 0.5;
```



A Scratch script consisting of a black hat with three colored buttons (red, yellow, green) at the top, followed by a script area containing the following code:

```
let probability = 0.5;
if (random(1) < probability) {
    //all random numbers between 0..0.49
    //...
} else {
    //all random numbers between 0.5..0.99
    //...
}
```

Putting it all together



```
function draw() {
  // Go row by row
  for (let y = 0; y < width; y += SPACING) {
    for (let x = 0; x < height; x += SPACING) {
      // Switch which "character"
      if (random(1) < PROBABILITY) {
        line(x, y, x + SPACING, y + SPACING);
      } else {
        line(x, y + SPACING, x + SPACING, y);
      }
    }
  }
}
```

<https://editor.p5js.org/legie/sketches/VDfxWIEAL>

What's Next?

Next

- What is that we are doing...?

→ Algorithmic Thinking

- How do we think about those tasks?

→ Languages & Environments

- And with what tools / systems...?

→ html, css, js, webserver

The End