

Creative Coding I - Session 02

MA Creative Technologies - WS '25'26



FILMUNIVERSITÄT
BABELSBERG
KONRAD WOLF

Jonathan Ho

Creative Technologies

Emergence



[B. Yirka. 2015. Researchers use wave theories to understand flocks of birds. Sign of the times.]

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

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.]

01.

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
```

Today

01 Emergence - con't

What is it?

02 Implementing 10 PRINT in p5.js

Loops, primitives, color

03 Emergence in the Wild

Reddit r/place experiment

04 Tools and Environments

Programming Languages, Frameworks, Libraries Software Hardware

05 Algorithmic Thinking

06 Instructions

Implementing 10 PRINT

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

Algorithm:

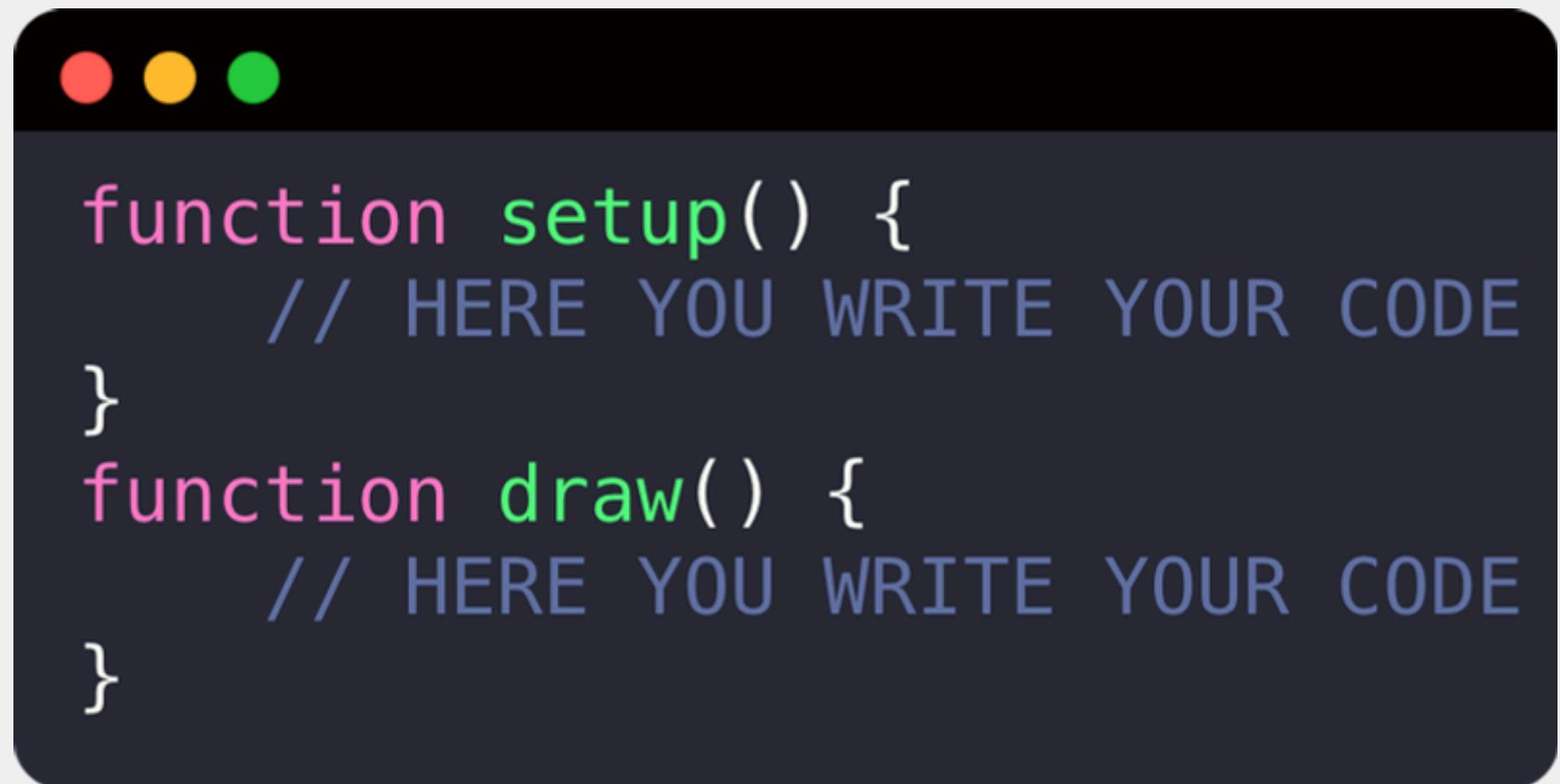
- Loop row by row
- Place a / or a \...
- ...randomly
-



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 (red, yellow, green) in the top-left corner. The code area contains the following pseudocode:

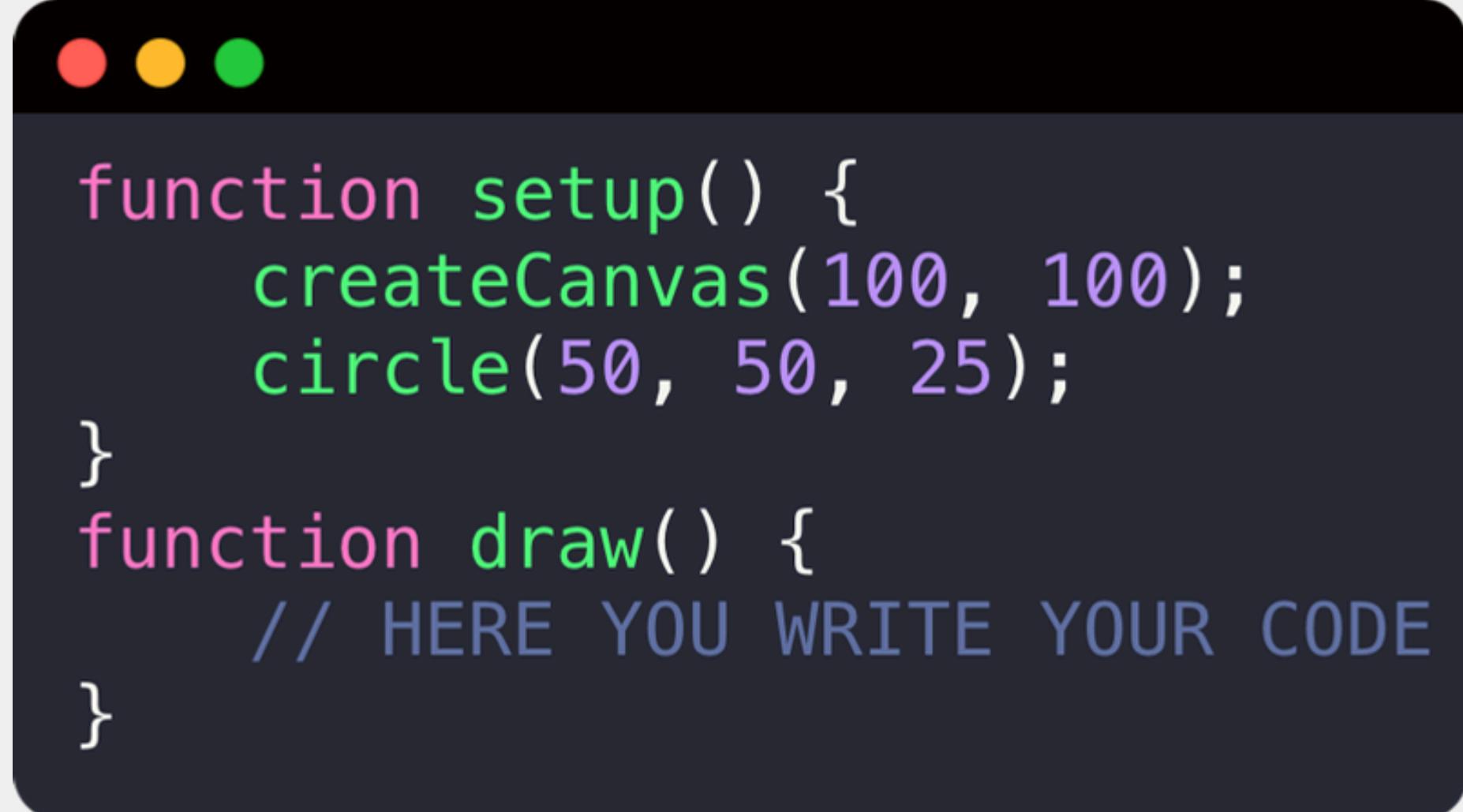
```
function setup() {  
    // HERE YOU WRITE YOUR CODE  
}  
function draw() {  
    // HERE YOU WRITE YOUR CODE  
}
```

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



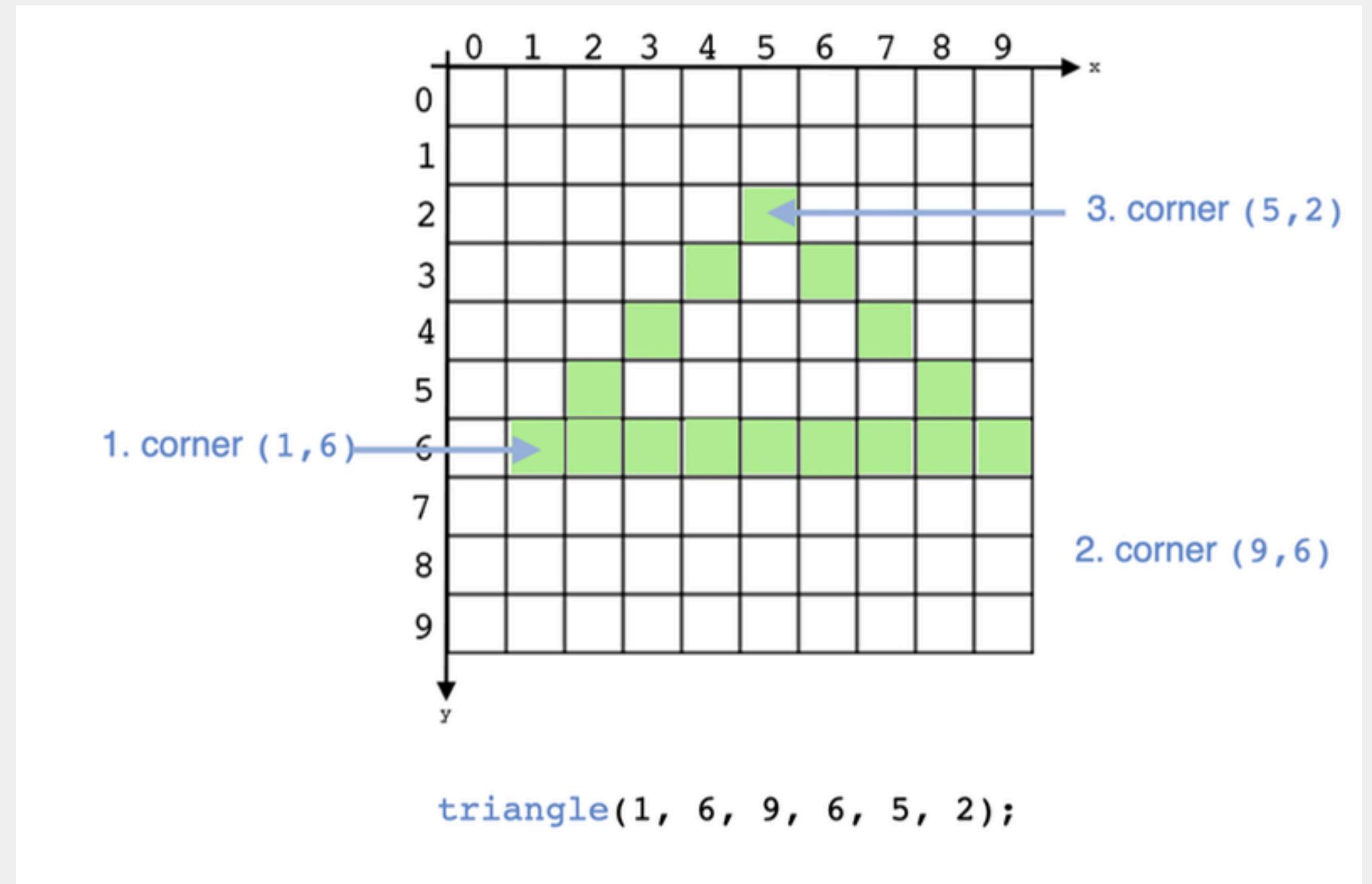
```
function setup() {
  createCanvas(100, 100);
  circle(50, 50, 25);
}

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)



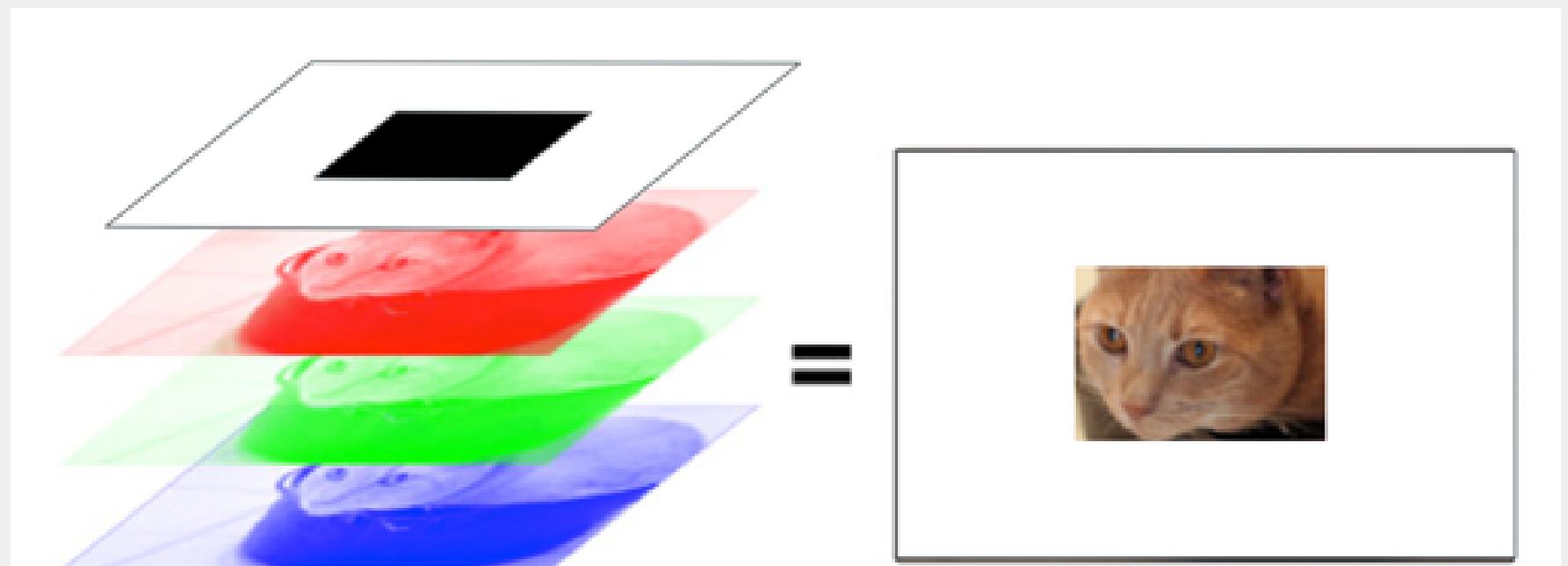
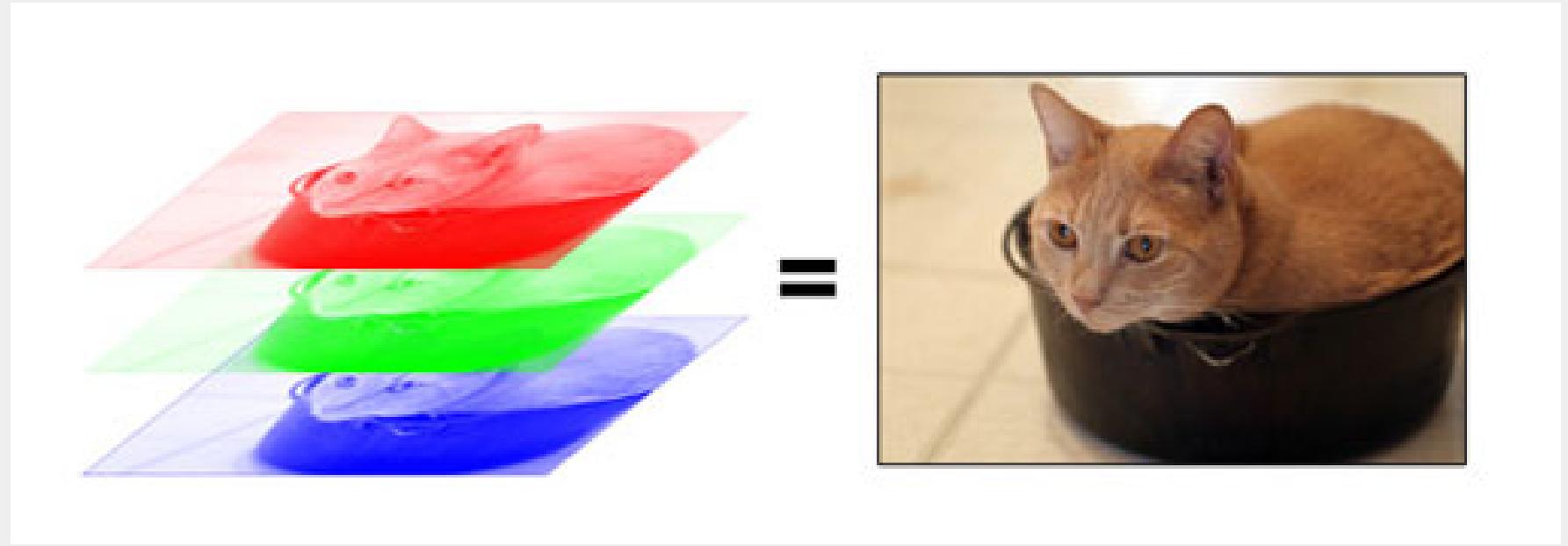
2D Primitives

- `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



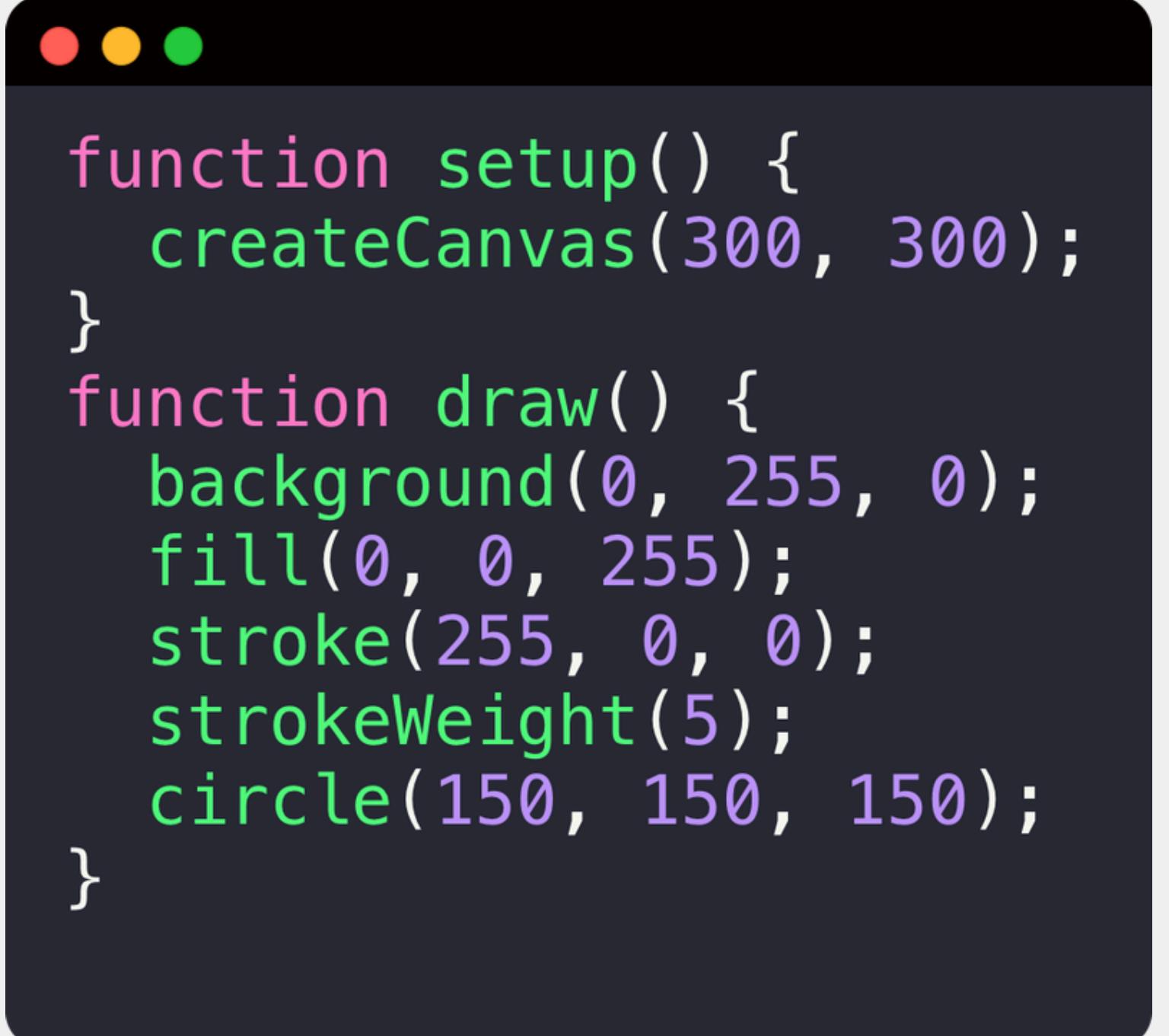
Color Function Calls

- 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();
```



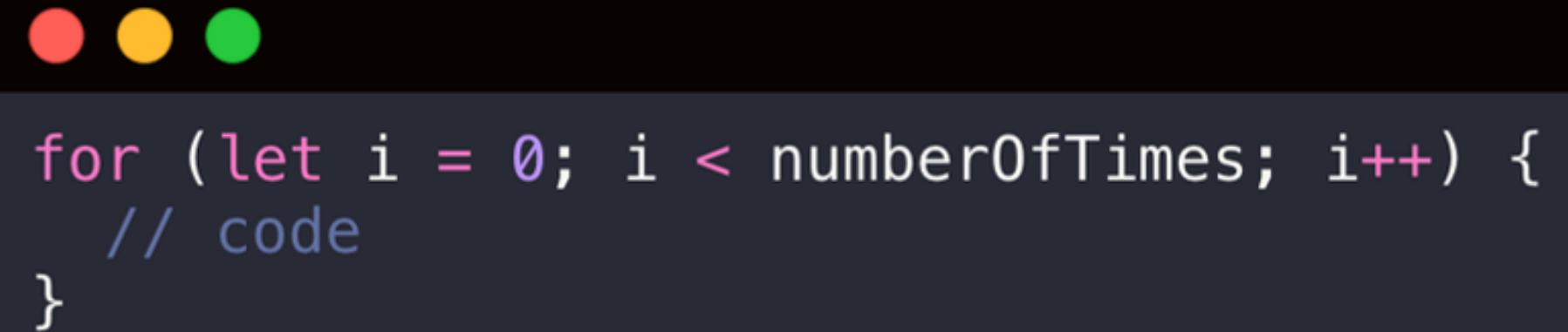
```
function setup() {
  createCanvas(300, 300);
}

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

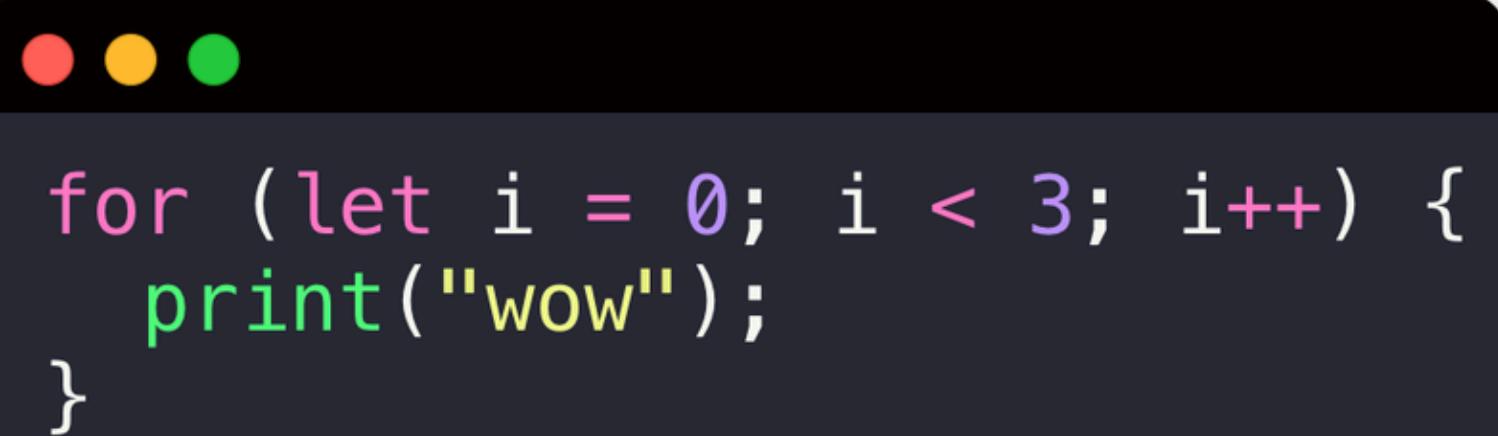


Loops

- **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



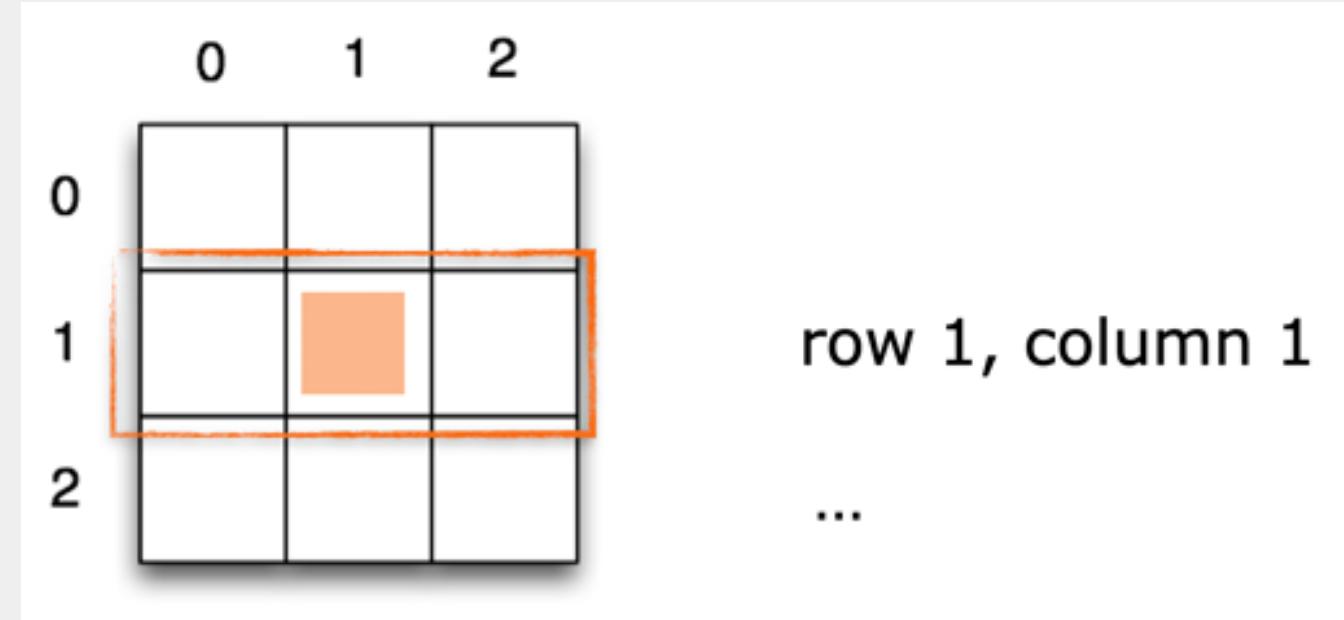
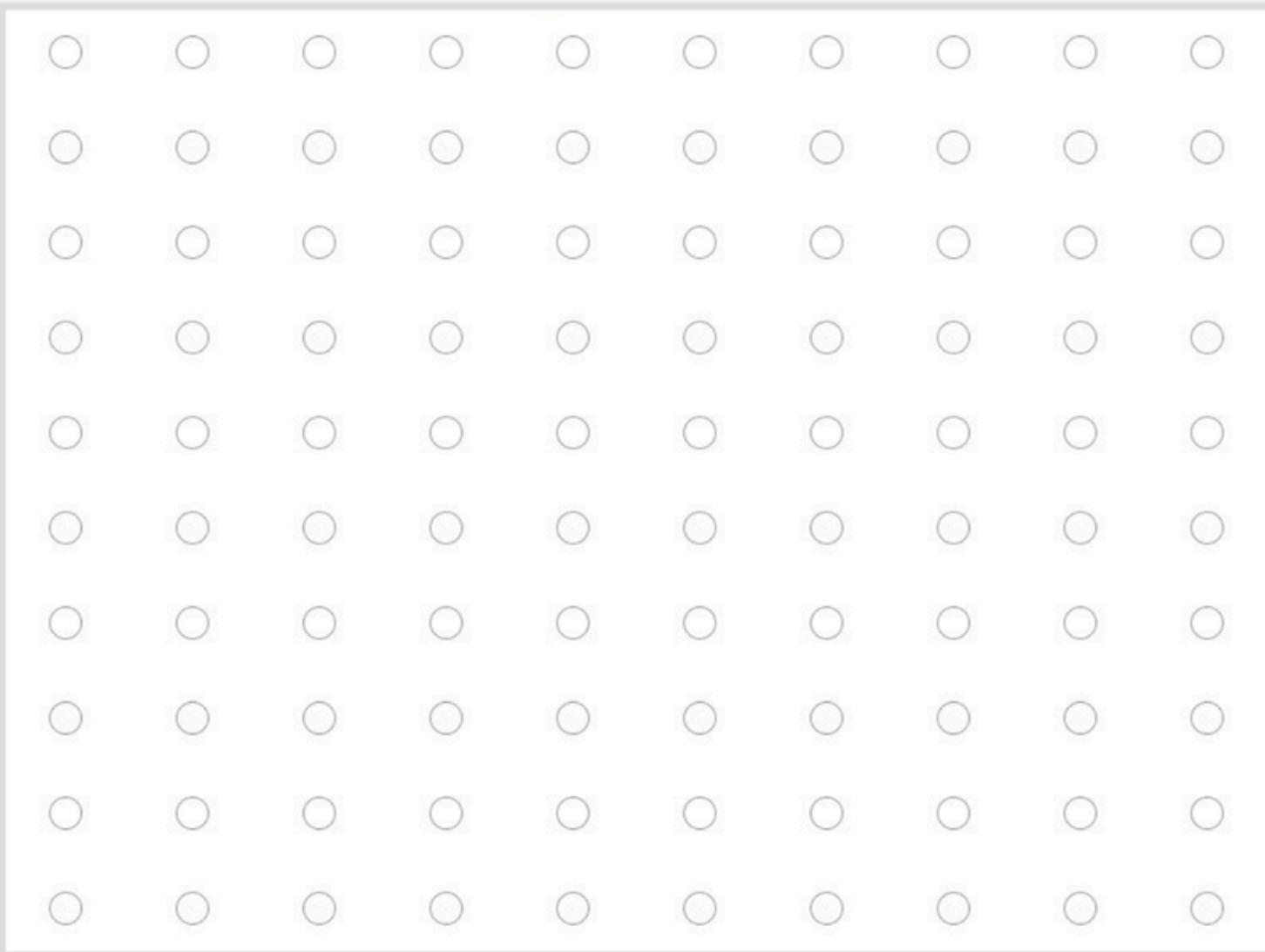
```
for (let i = 0; i < number0fTimes; i++) {  
    // code  
}
```



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

Example

Loops

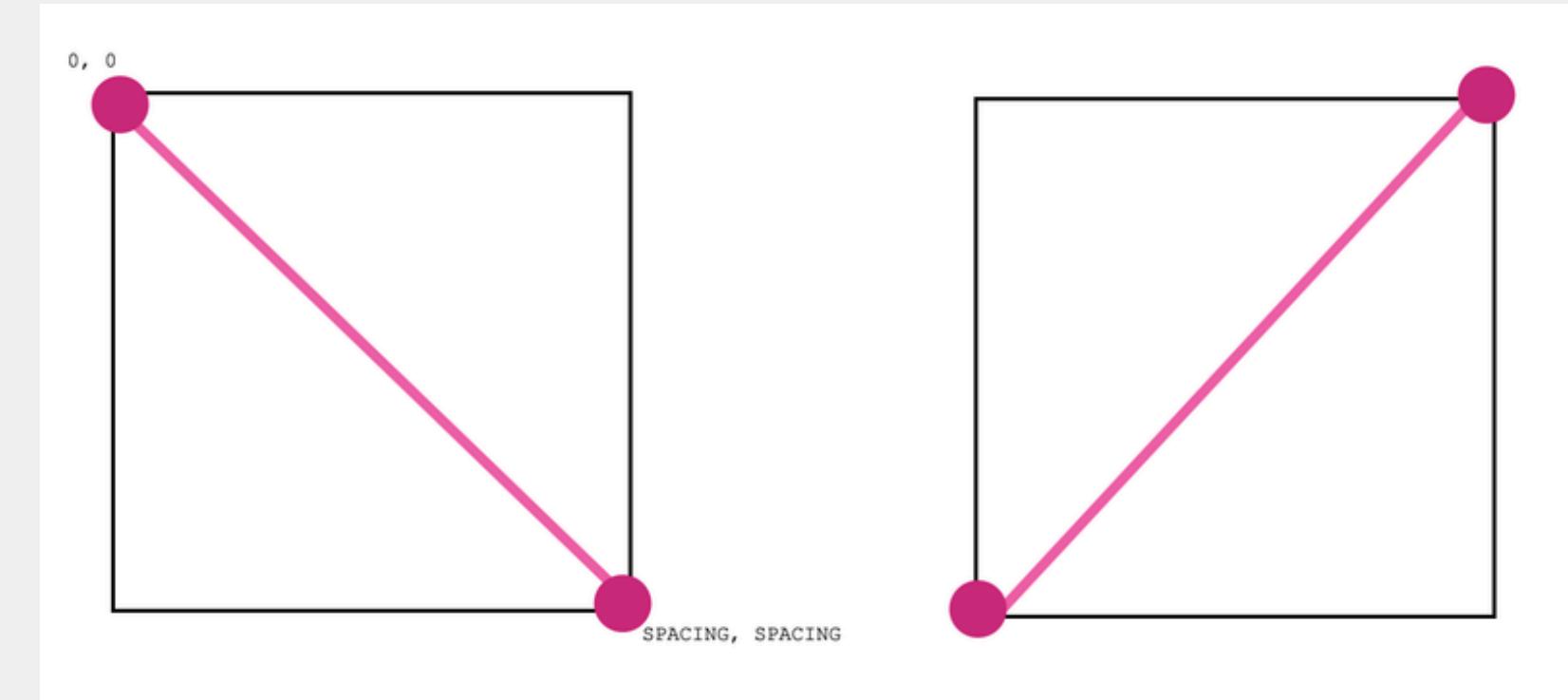


// Pseudo Code
For every row {
 For every column {
 }
}

The slide features a dark-themed window with red, yellow, and green circular icons at the top. Inside, there is a block of pseudo code:

Implementing 10 PRINT

- How would we place a “ / ” or a “ \ ” ? on each space?



Algorithm:

- Draw a line
 - starting at x, y
 - ending at $x + 1$ space or $y + 1$



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

Implementing 10 PRINT

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

Algorithm:

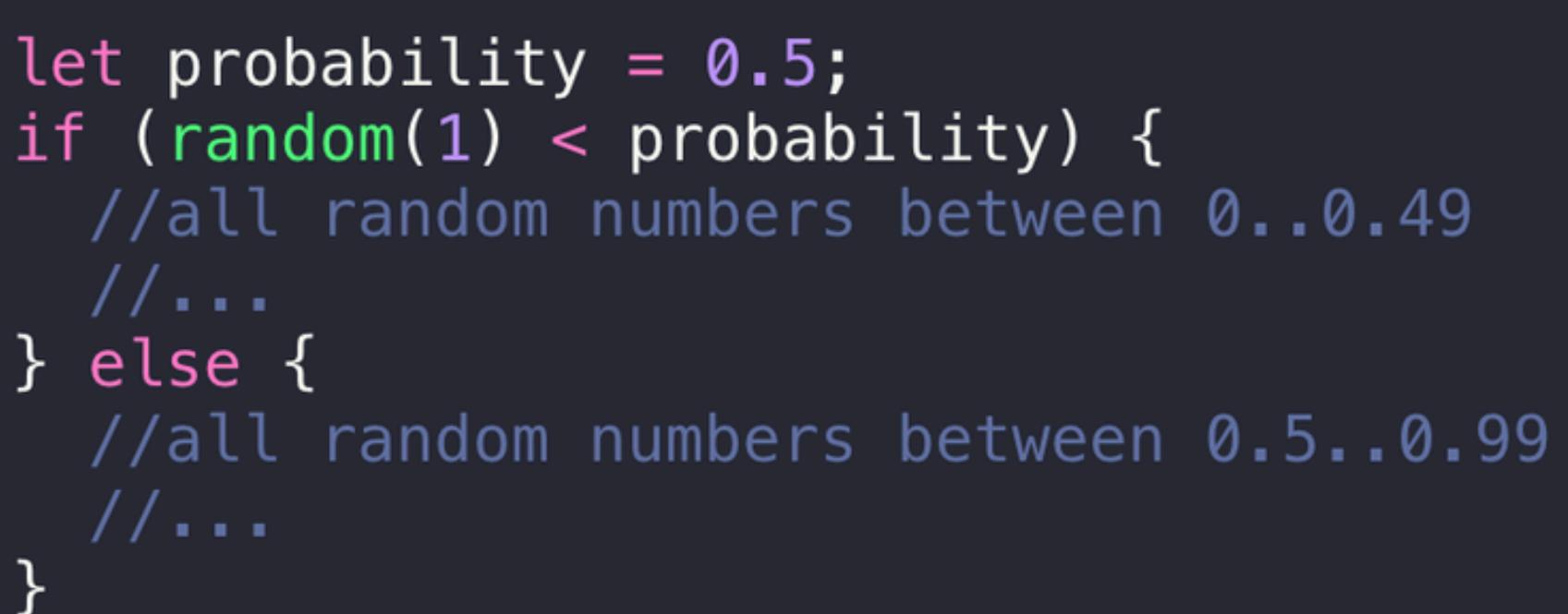
- Go row by row ✓
- Place a / or a \... ✓
- ...randomly ??
-



Randomness

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

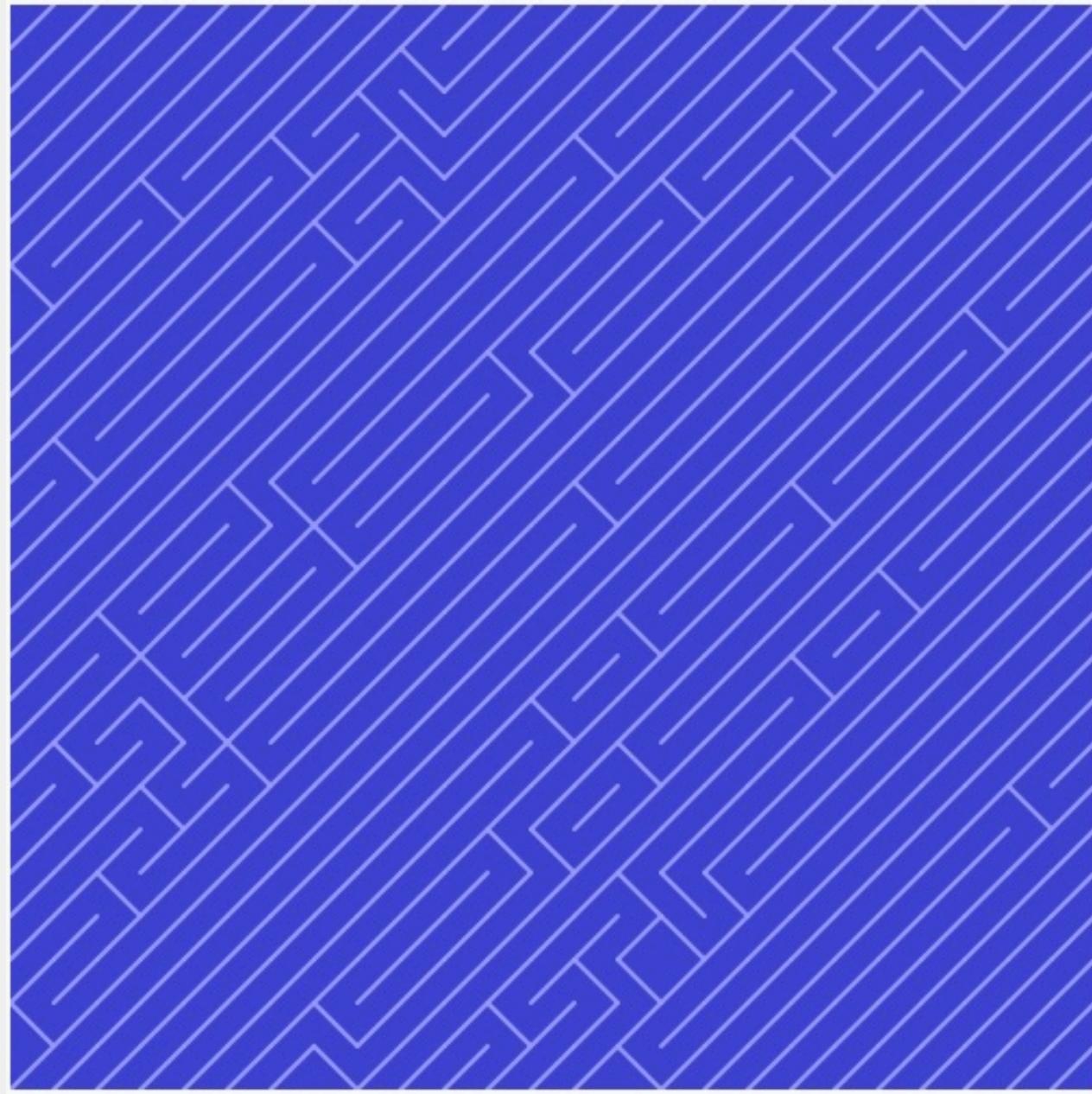
```
let probability = 0.5;
```



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

A Scratch script with three colored circles at the top. It contains a conditional loop. The condition checks if a random number (random(1)) is less than the variable probability (0.5). If true, it executes the code block below, which includes a comment about generating numbers between 0..0.49 and a placeholder for more code. If false, it executes the code block for the else branch, which includes a comment about generating numbers between 0.5..0.99 and another placeholder for more code.

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>

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

??

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

Last Week | Emergence

Reddit's r/place

6 / 92

Reddit's r/place

There is an empty canvas. You may place a tile upon it, but you must wait to place another. Individually you can create something. Together you can create something more.

[J. Ho. 2022. Our Place: On the Emergence of Visual Artifacts in Collaborative Art in Reddit's r/place
. Academic Conference Simulation at FUB 2022.]

Reddit's r/place

There is an empty canvas. You may place a tile upon it, but you must wait to place another. Individually you can create something. Together you can create something more.



6 / 92

[J. Ho. 2022. Our Place: On the Emergence of Visual Artifacts in Collaborative Art in Reddit's r/place
. Academic Conference Simulation at FUB 2022.]



Last Week | Emergence

Reddit's r/place

The communal co-creation of an artistic artifact.

[J. Ho. 2022. Our Place: On the Emergence of Visual Artifacts in Collaborative Art in Reddit's r/place
. Academic Conference Simulation at FUB 2022.]

7 / 92

Reddit's r/place

The communal co-creation of an artistic artifact.

- **72-hour social experiment, 2017 & 2022**

[J. Ho. 2022. Our Place: On the Emergence of Visual Artifacts in Collaborative Art in Reddit's r/place
. Academic Conference Simulation at FUB 2022.]

Reddit's r/place

The communal co-creation of an artistic artifact.

- 72-hour social experiment, 2017 &
- 2022 1000 x 1000-pixel canvas

[J. Ho. 2022. Our Place: On the Emergence of Visual Artifacts in Collaborative Art in Reddit's r/place
. Academic Conference Simulation at FUB 2022.]

Reddit's r/place

The communal co-creation of an artistic artifact.

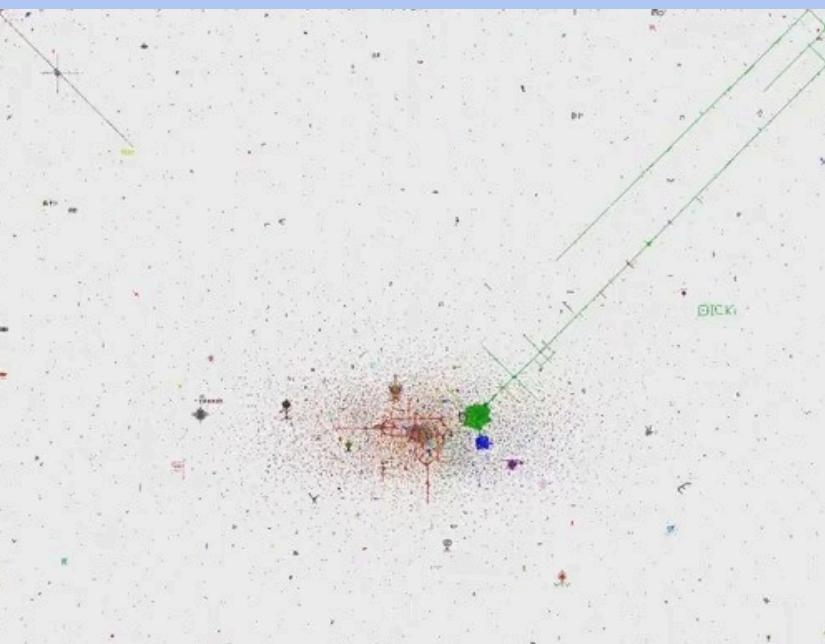
72-hour social experiment, 2017 & 2022
1000 x 1000-pixel canvas Set the color of a single pixel, every 5 to 20 minutes

[J. Ho. 2022. Our Place: On the Emergence of Visual Artifacts in Collaborative Art in Reddit's r/place . Academic Conference Simulation at FUB 2022.]

7 / 92

Last Session | Emergence

Reddit's r/place



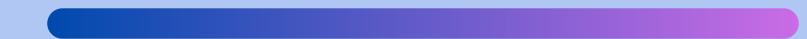
[J. Ho. 2022. Our Place: On the Emergence of Visual Artifacts in Collaborative Art in Reddit's r/place
. Academic Conference Simulation at FUB 2022.]

8 / 92

Reddit's r/place

- **Users had to collaborate to create larger artifacts**
 - **Subreddits, interest-based subforums on Reddit**
 - **Discord channels**

[J. Ho. 2022. Our Place: On the Emergence of Visual Artifacts in Collaborative Art in Reddit's r/place
. Academic Conference Simulation at FUB 2022.]



Reddit's r/place

- **Users had to collaborate to create larger artifacts**
 - Subreddits, interest-based subforums on Reddit
 - Discord channels
- **On-going battles and coalition-forming regarding space**

[J. Ho. 2022. Our Place: On the Emergence of Visual Artifacts in Collaborative Art in Reddit's r/place

. Academic Conference Simulation at FUB 2022.]

Last Week | Emergence

Reddit's r/place



[J. Ho. 2022. Our Place: On the Emergence of Visual Artifacts in Collaborative Art in Reddit's r/place
. Academic Conference Simulation at FUB 2022.]

10 / 92

Last Week | Emergence

Reddit's r/place

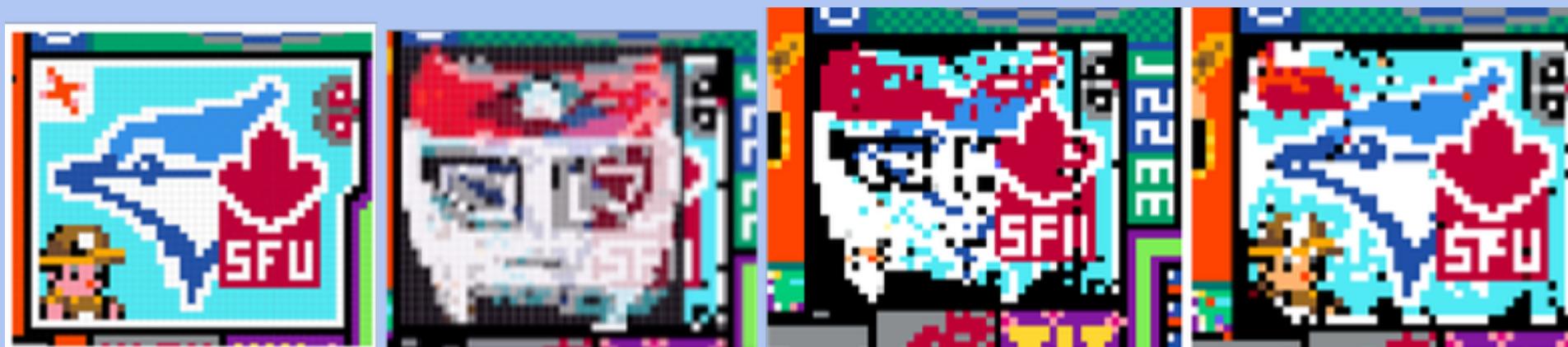


[J. Ho. 2022. Our Place: On the Emergence of Visual Artifacts in Collaborative Art in Reddit's r/place
. Academic Conference Simulation at FUB 2022.]

11 / 92

Last Week | Emergence

Reddit's r/place



[J. Ho. 2022. Our Place: On the Emergence of Visual Artifacts in Collaborative Art in Reddit's r/place
. Academic Conference Simulation at FUB 2022.]

12 / 92

Last Week | Emergence

Reddit's r/place

13 / 92

Last Week | Emergence

Reddit's r/place

What were the driving forces behind the creation of coherent artworks and visual artifacts?

[K.T. Litherland and A. I. Mørch. 2021. Instruction vs. emergence on r/place: Understanding the growth and control of evolving artifacts in mass collaboration. *Computers in Human Behavior*][J. Ho. 2022. Our Place: On the Emergence of Visual Artifacts in Collaborative Art in Reddit's r/place. Academic Conference Simulation at FUB 2022.]

13 / 92

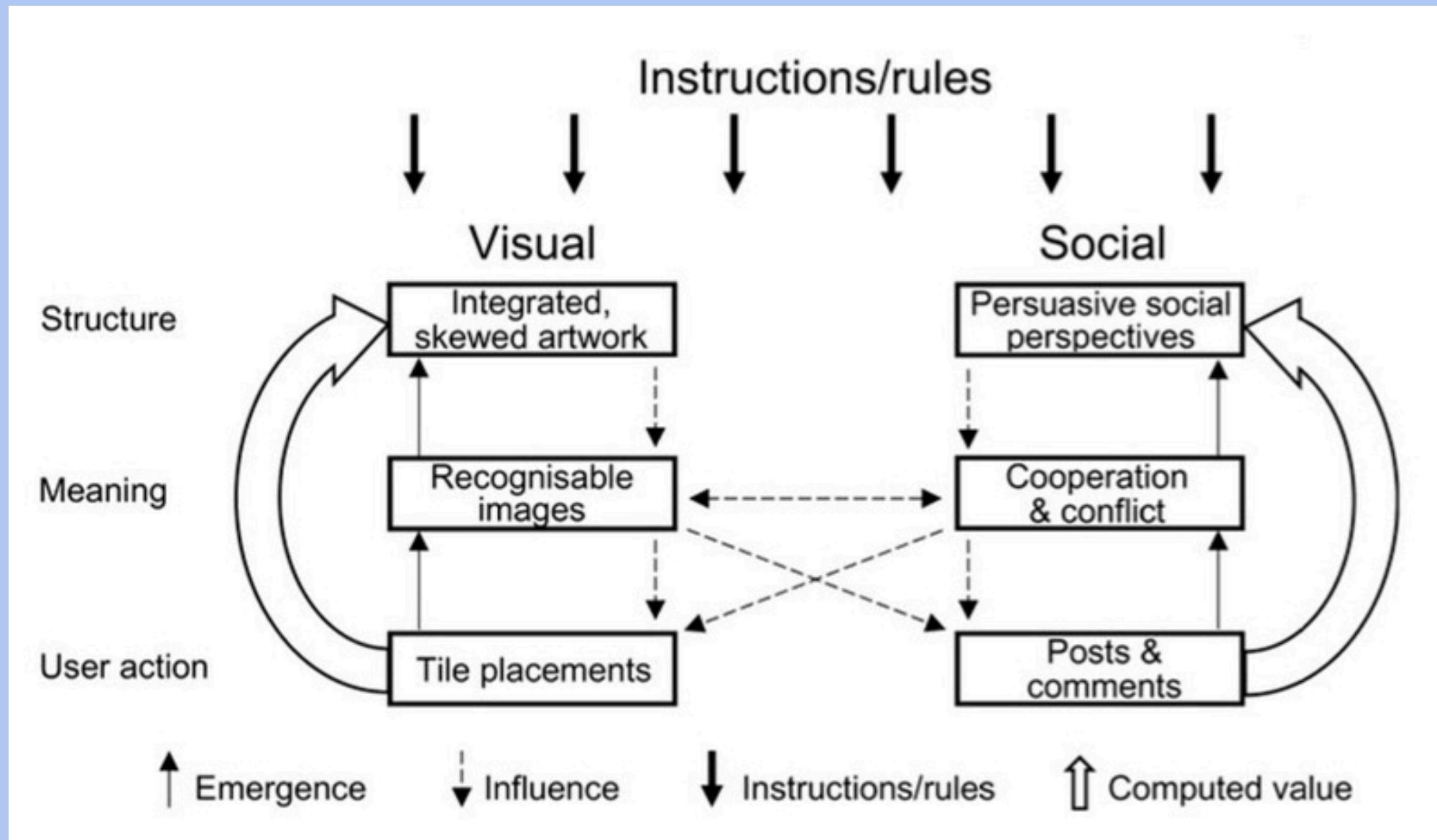
Reddit's r/place

What were the driving forces behind the creation of coherent artworks and visual artifacts?

K.T. Litherland and A. I. Mørch:

Instructions (top down) vs. Emergence (bottom up)

[K.T. Litherland and A. I. Mørch. 2021. Instruction vs. emergence on r/place: Understanding the growth and control of evolving artifacts in mass collaboration. Computers in Human Behavior] [J. Ho. 2022. Our Place: On the Emergence of Visual Artifacts in Collaborative Art in Reddit's r/place. Academic Conference Simulation at FUB 2022.]



[K.T. Litherland and A. I. Mørch. 2021. Instruction vs. emergence on r/place: Understanding the growth and control of evolving artifacts in mass collaboration. *Computers in Human Behavior*][J. Ho. 2022. Our Place: On the Emergence of Visual Artifacts in Collaborative Art in Reddit's r/place. Academic Conference Simulation at FUB 2022.]

14 / 92

Last Week

Emergence

CaseyReas - On Creative Coding and Emergence ↗

15 / 92

Last Week

Creative Coding

16 / 92

Last Week

Creative Coding

What do I have available and what can I do with that beyond the obvious?

16 / 92

Last Week

Creative Coding

What do I have available and what can I do with that beyond the obvious?

- Producing something expressive rather than focussing on a practical use
- Developing software beyond its standard usage scenarios
- Developing tools that help others to be creative

Creative Coding

Practical Approaches

17 / 92

Creative Coding

.Practical Approaches

Algorithms to create graphics and audio

17 / 92

Creative Coding

Practical Approaches

Algorithms to create graphics and audio

→ TBAG, Procedural Generation and Simulation

17 / 92

- **Practical Approaches**
- **Algorithms to create graphics and audio**
- **Smart, meaningful or unusual data sources**

Practical Approaches

- Algorithms to create graphics and audio
- Smart, meaningful or unusual data sources
 - Images, video, audio
 - Camera and microphone
 - Online resources such as Twitter, Instagram and ChatGPT
 - Mobile devices as sensors
 - ...

Practical Approaches

- Algorithms to create graphics and audio
- Smart, meaningful or unusual data sources
- Diverse output formats

Practical Approaches

- Algorithms to create graphics and audio
- Smart, meaningful or unusual data
- sources Diverse output formats
 - Web
 - From large-scale such as buildings to small-scale such as smart watches
 - Multi-screen setups for example with mobile devices
 - ...
 -

Homework

20 / 92

Instructions

21 / 92

Today

22 / 92

Today

Tools and Environments

22 / 92

Today

- **Tools and Environments**
- **Algorithmic Thinking**

22 / 92

Today

- **Tools and Environments**
- **Algorithmic Thinking**
- **Instructions**

Tools and Environments

Algorithmic Thinking
Artistic Interpretation

23 / 92

Tools and Environments

24 / 92

.Tools and Environments

Programming Languages, Frameworks, Libraries

24 / 92

.Tools and Environments

- **Programming Languages, Frameworks,
Libraries Software**

.Tools and Environments

- Programming Languages, Frameworks,
- Libraries Software Hardware

What Are Programming Languages?

25 / 92

Wikipedia says...

A programming language is a formal language, which comprises a set of instructions that produce various kinds of output.

Wikipedia says...

A programming language is a formal language, which comprises a set of instructions that produce various kinds of output.

Programming languages are used in computer programming to implement **algorithms**.

Wikipedia says...

A programming language is a formal language, which comprises a set of instructions that produce various kinds of output.

Programming languages are used in computer programming to implement **algorithms**.

A programming language's surface form is known as its syntax.

Tools and Environments | Programming Languages

Programming

You have to learn

27 / 92

Programming

You have to learn

1. The formal language

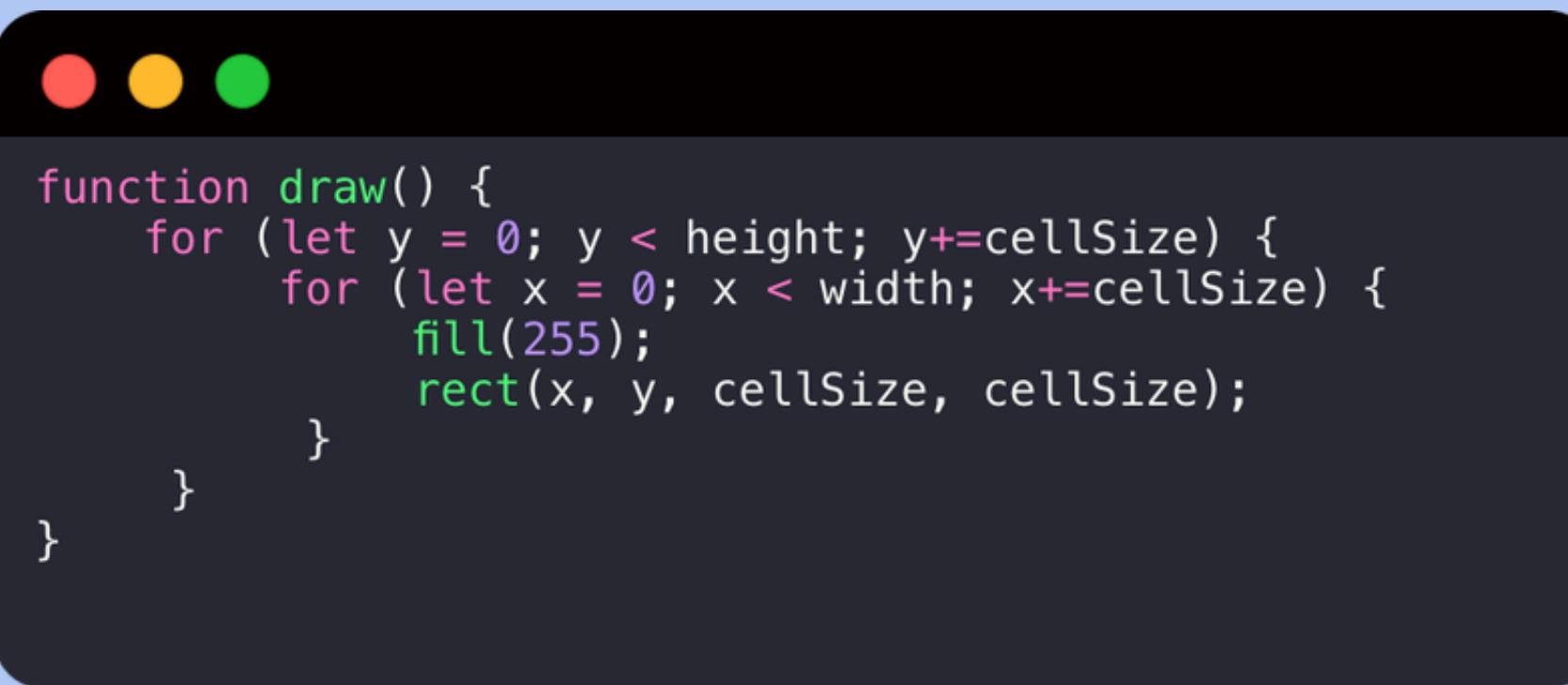
2. Algorithm

Tools and Environments | Programming Languages | Programming

Examples

28 / 92

Examples



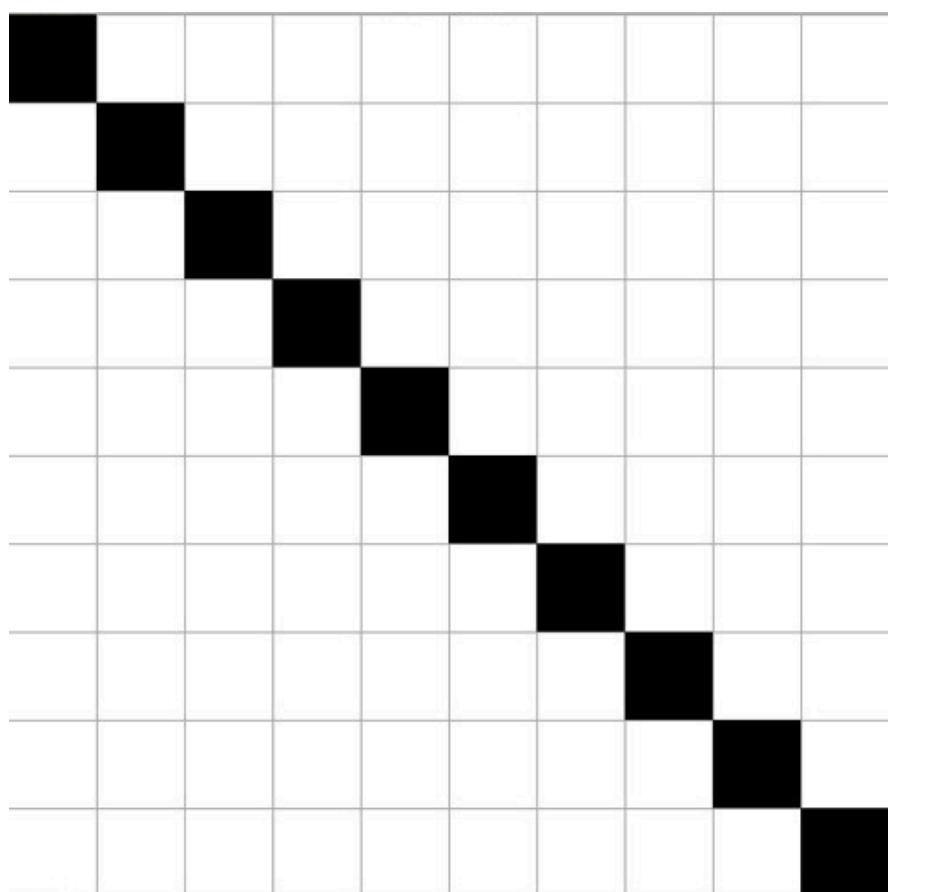
A screenshot of a code editor window titled "Sketch". The window has three colored window control buttons (red, yellow, green) at the top. The code area contains the following JavaScript-like pseudocode:

```
function draw() {
    for (let y = 0; y < height; y+=cellSize) {
        for (let x = 0; x < width; x+=cellSize) {
            fill(255);
            rect(x, y, cellSize, cellSize);
        }
    }
}
```

Examples

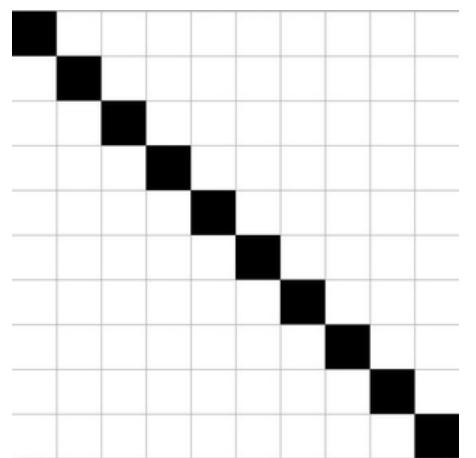
How do you need to control the fill command to create the following examples?

Examples



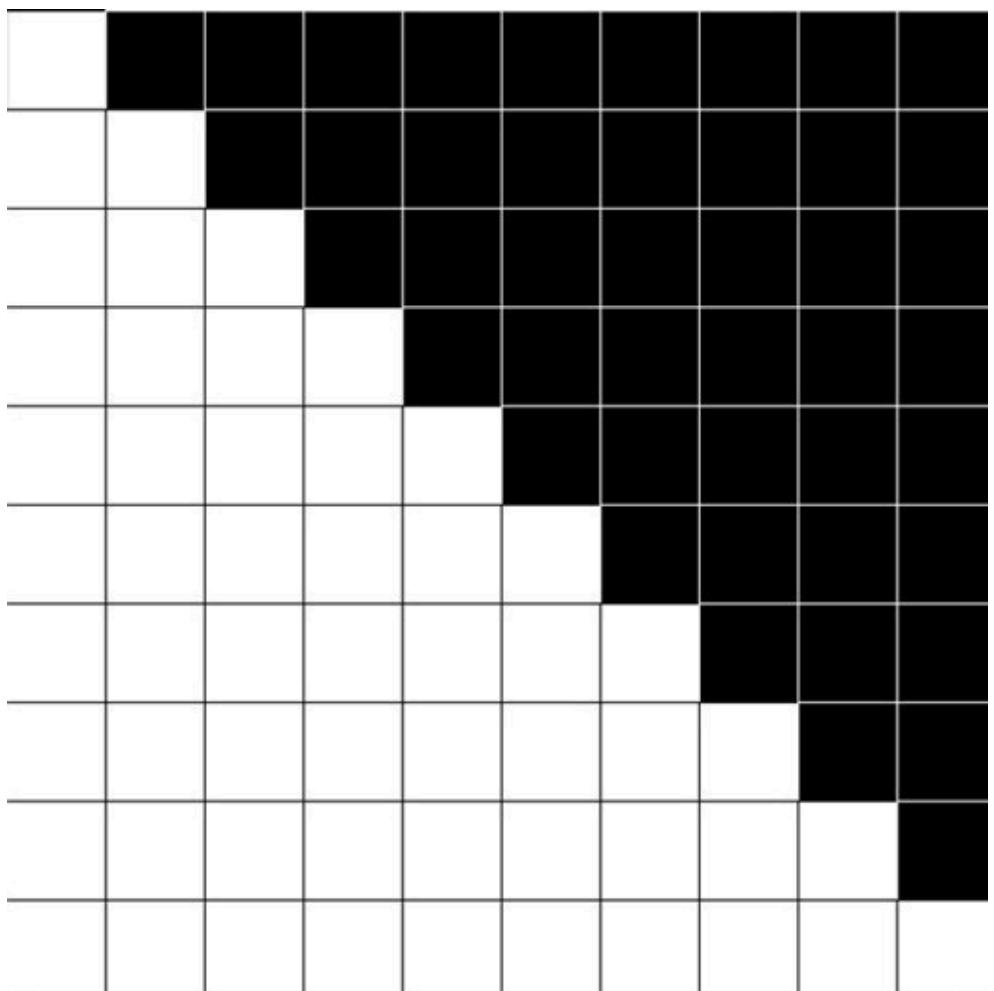
30 / 92

Examples



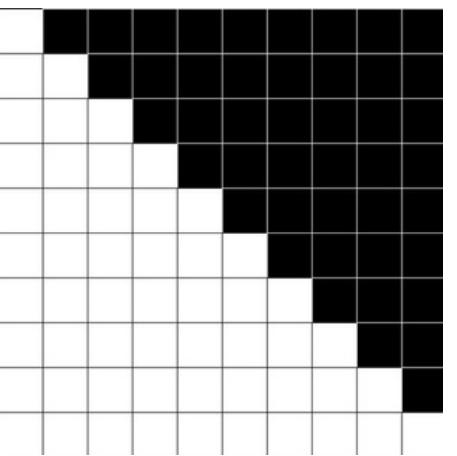
```
//  
// https://editor.p5js.org/legie/sketches/IWJGIhhtI  
// Grid Examples  
  
...  
  
fill(255);  
// Fill only the cell  
// on the diagonal  
if ( y == x ) {  
    fill(0);  
}  
rect(x, y, cellSize, cellSize);
```

Examples



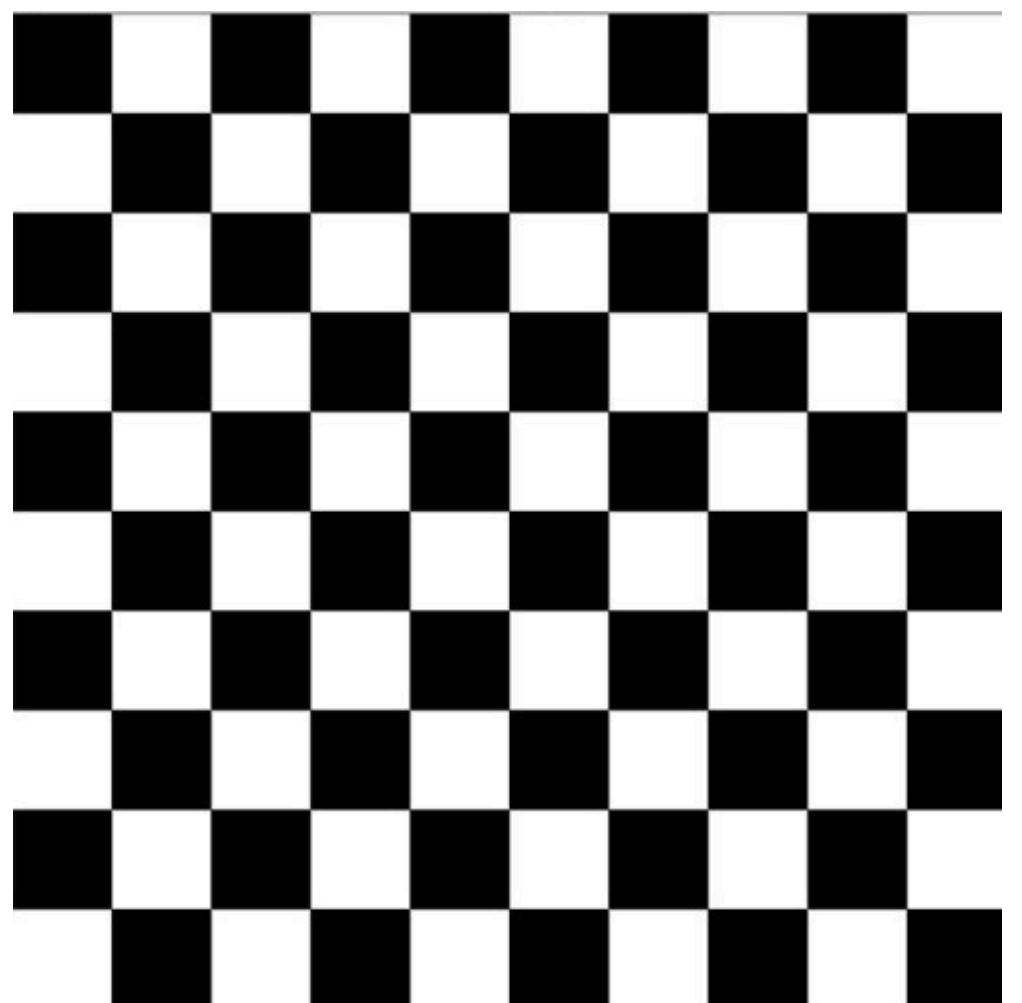
32 / 92

Examples



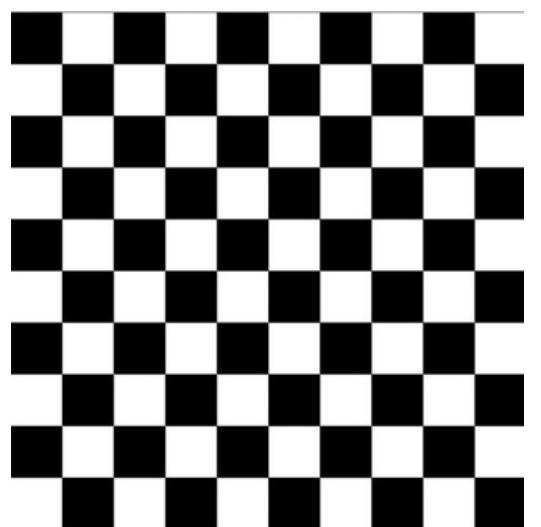
```
//  
// https://editor.p5js.org/legie/sketches/5x1bAs66K  
// Grid Examples  
  
... stroke(0);  
fill(255);  
  
if (x > y) {  
stroke(255);  
fill(0);  
}  
rect(x, y, cellSize, cellSize);
```

Examples



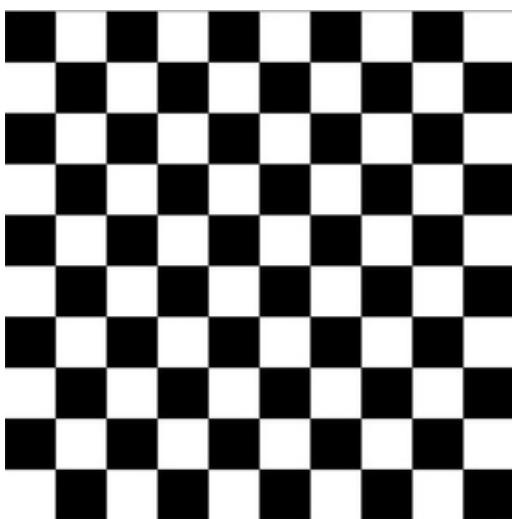
Tools and Environments | Programming Languages | Programming

Examples



35 / 92

Examples



```
// https://editor.p5js.org/legie/sketches/_NHk4arDR
// Grid Examples
```

```
if ((row % 2 == 0) && (column % 2 == 0)) {
    fill(0);
} else if ((row % 2 != 0) && (column % 2 != 0)) {
    fill(0);
}
```

Tools and Environments | Programming Languages | Programming

The Modulo Operator

37 / 92

The Modulo Operator

The modulo operator returns for a division with a whole number the rest of that division:

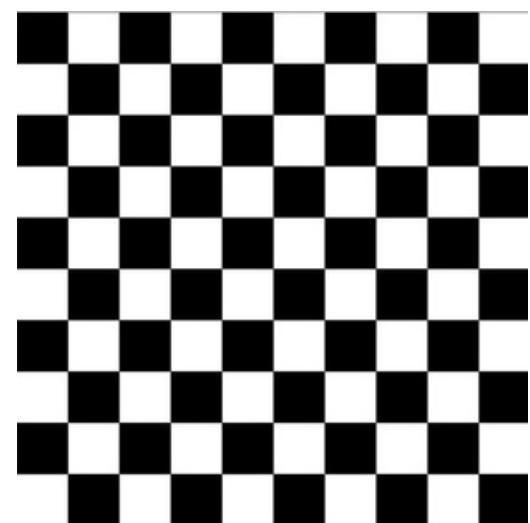
```
// Pseudo Code 5 / 2 is 2 (the  
quotient) with rest 1  
7 / 5 is 1 with rest 2  
x / y is quotient q with rest r  
x = q * y + r
```

The Modulo Operator

This comes in handy when testing for even numbers:

```
if(number % 2 == 0){  
    print("even");  
}
```

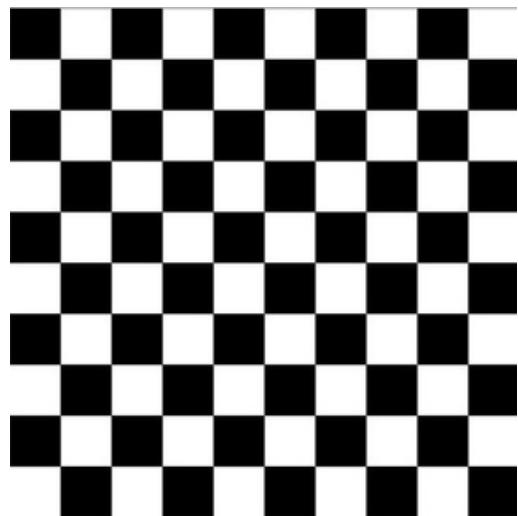
Examples



```
// https://editor.p5js.org/legie/sketches/_NHk4arDR
// Grid Examples
```

```
if((row + column) % 2 == 0) {
    fill(0);
}
```

Examples



```
if((row + column)% 2 == 0){  
    fill(0);  
}
```

- number sequence: a, b, c,
- d b and d are odd a is the
- even number before b c is
- the even number before d

$$b + d = (a+1) + (c+1) = a + c + 2$$

Programming

You have to learn

1. The formal language

2. Algorithm

Programming

You have to learn

1. The formal language

2. Algorithm

Why should you learn programming?