

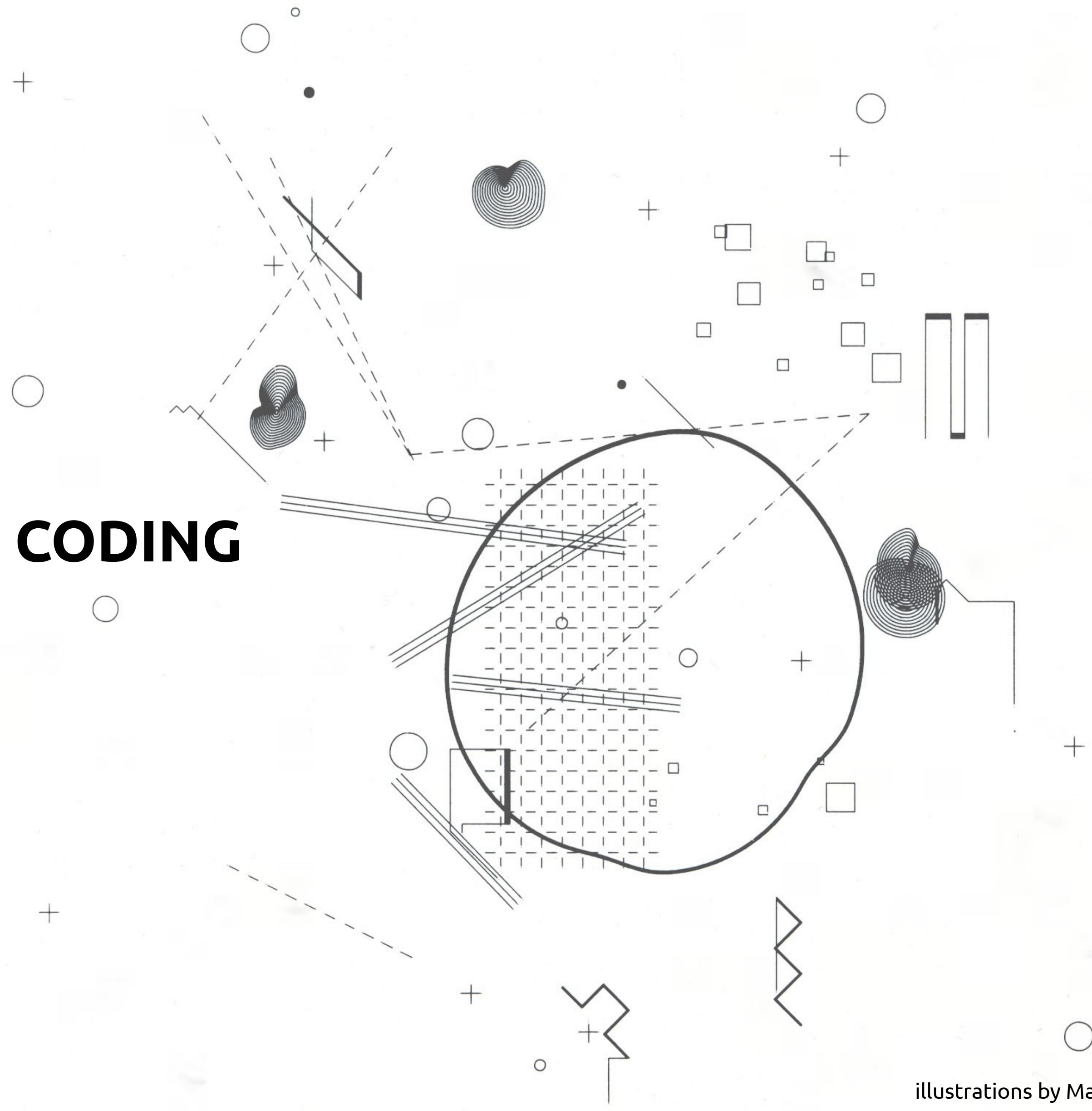
# INTRODUCTION TO GRAPHICS CODING

Term 1

November 2019

**Lior Ben-Gai**

<https://soogbet.github.io/ShenkarCC/>



illustrations by Manfred Mohr

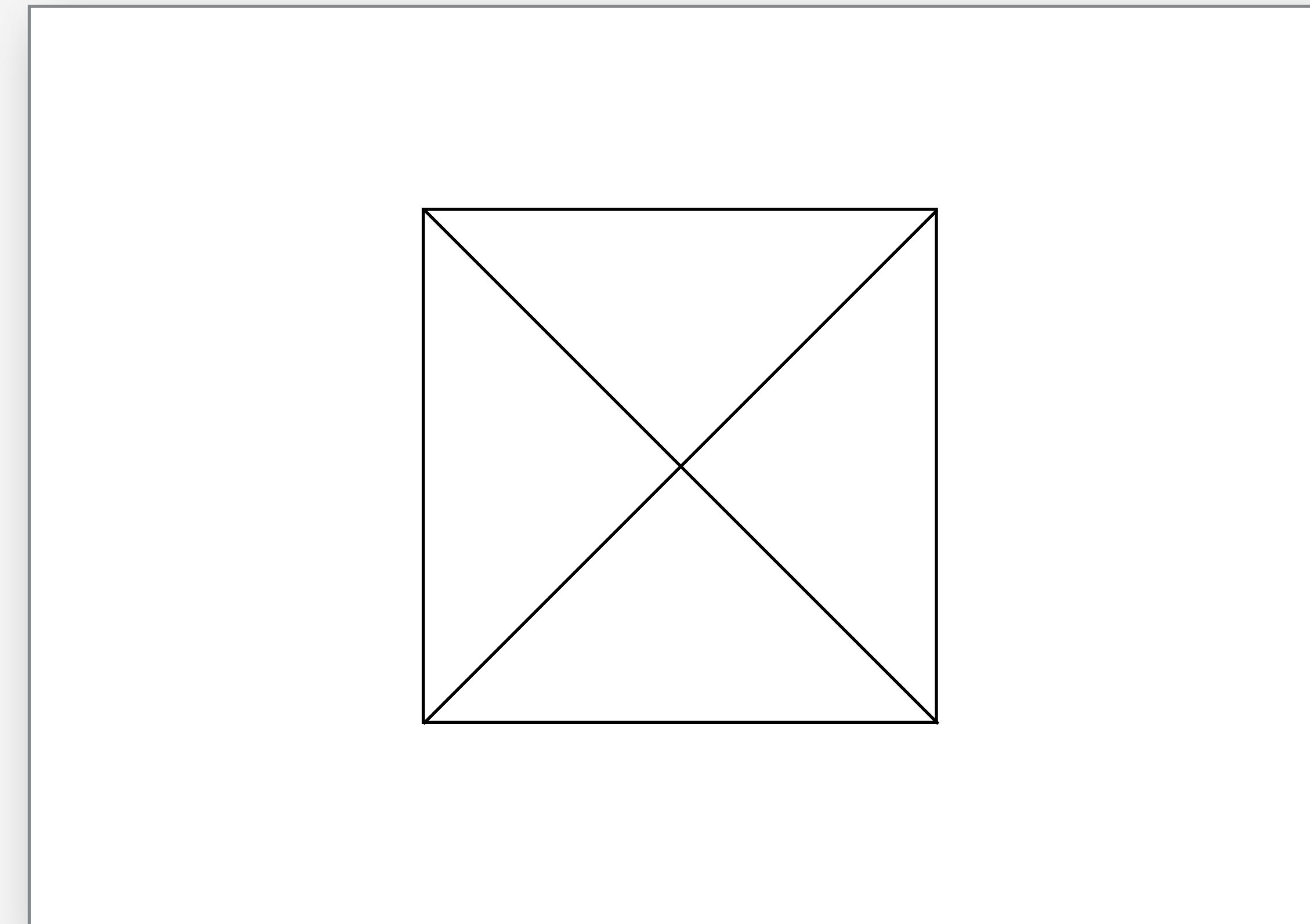
## GOALS

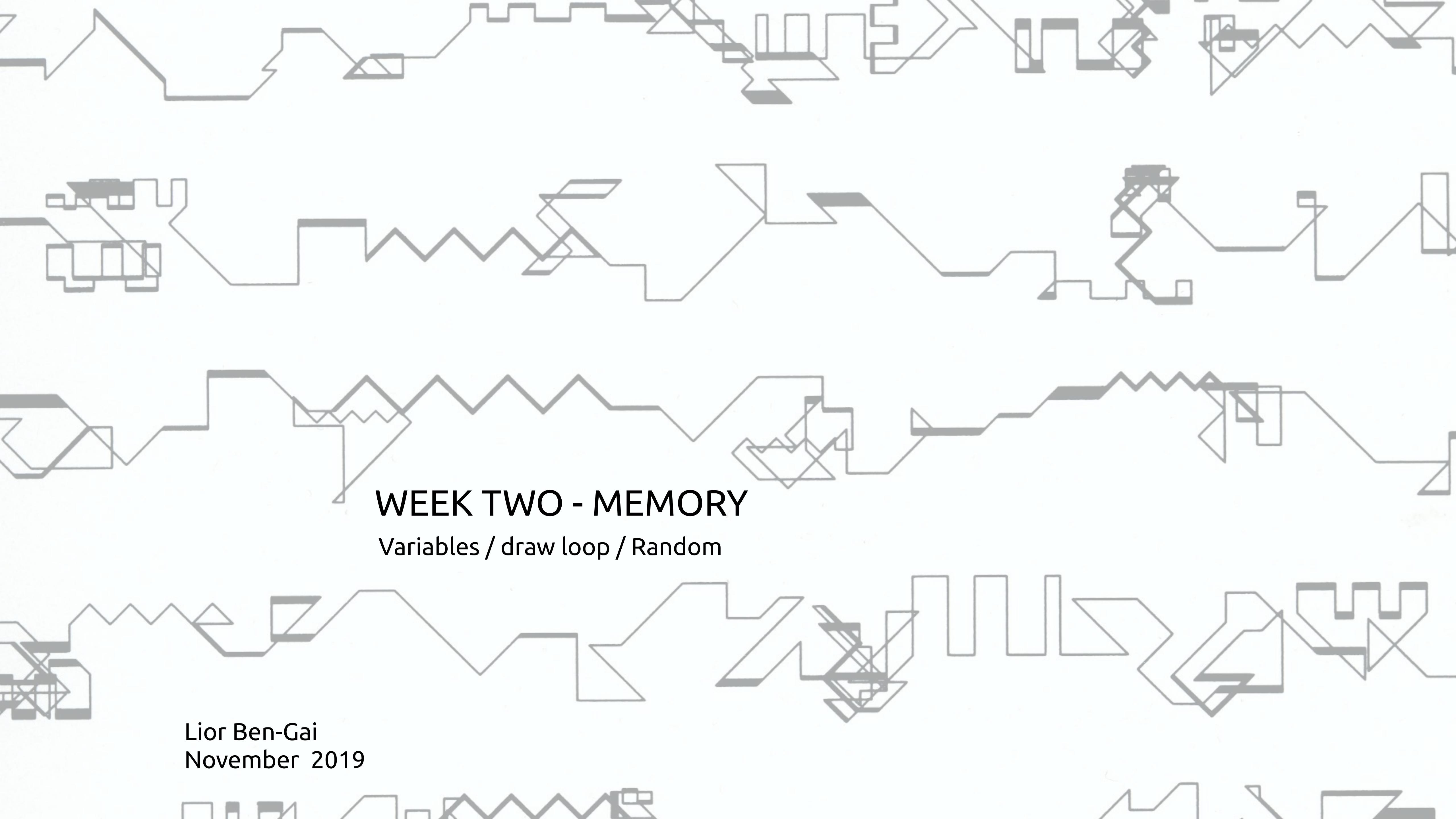
- 1. Improve our visual communication skills through computer programming.**
- 2. Improve our computer programming skills through visual communication.**

## COURSE TOPICS

1. Language, abstraction and procedural drawing
2. Variables, Functions, Conditions and Loops
3. Interaction and animation
4. Generative design methods
5. Data structures and visualization
6. Introduction to 3D structures, image processing and HTML

# DRAWING LANGUAGE





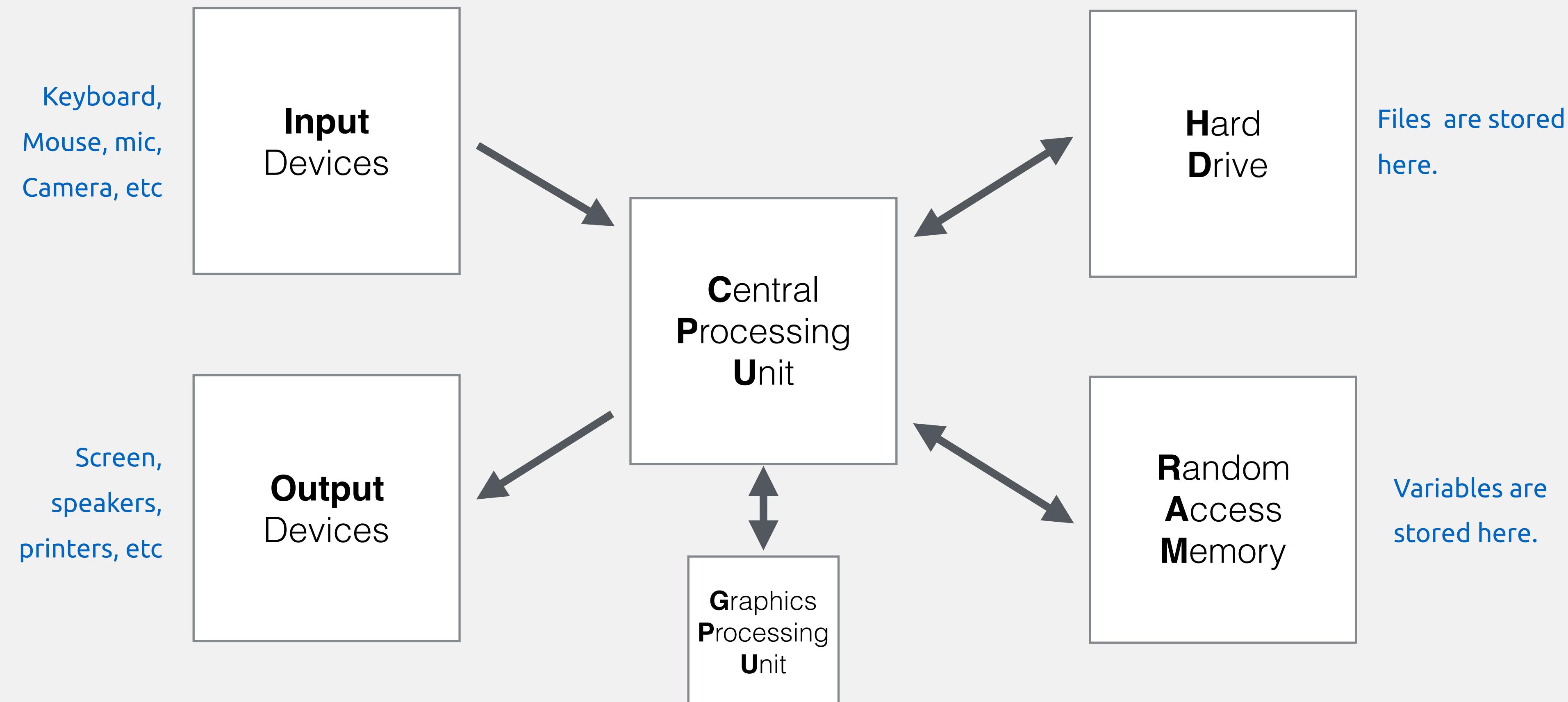
## WEEK TWO - MEMORY

Variables / draw loop / Random

Lior Ben-Gai  
November 2019

# HIGH LEVEL VIEW OF A COMPUTER

(very high level)



# DEFINE A VARIABLE

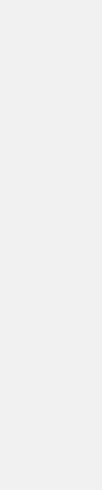
and assign it with:

```
var counter = 0;
```

Please define



A place in RAM named:



A value



# WEEK THREE - DECISION

Conditionals / Console / User input

Lior Ben-Gai  
November 2019

## DECISION

```
( ( 1 == 1 ) && ( 1 == 2 ) )
```



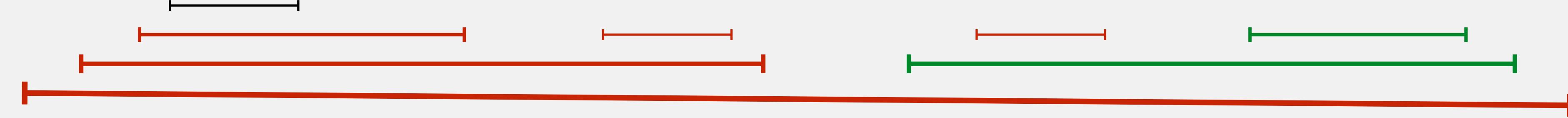
false

**AND** operator ( **&&** ) requires both sides to be **true**

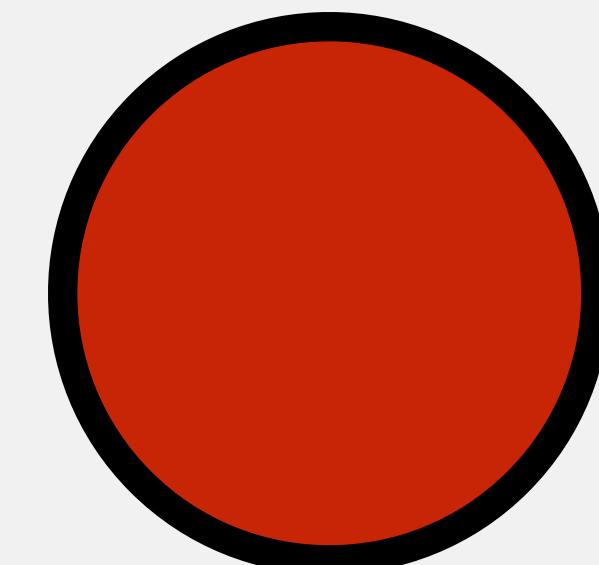
# EXERCISE

```
var flagA = true;  
var flagB = false;  
var flagC = ( flagA && flagB );
```

```
var flagD = ( ( ( 1 + 1 == 3 ) || flagC ) && ( (!true) || ( 2 >= 2 ) ) );
```



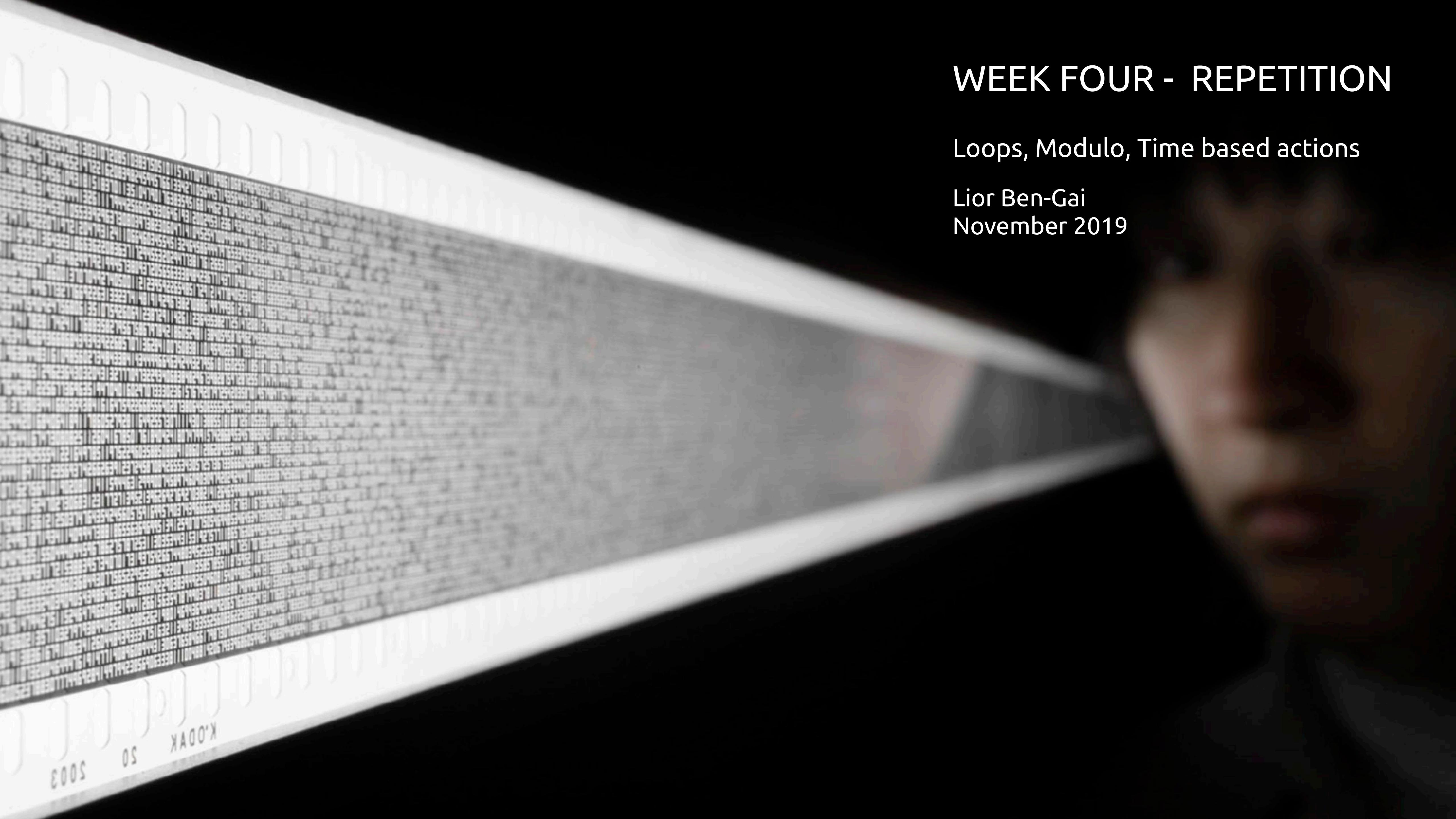
```
if(flagD){  
    fill(0,255,0);  
} else {  
    fill(255,0,0);  
}  
ellipse(530, 720, 200, 200);
```



# WEEK FOUR - REPETITION

Loops, Modulo, Time based actions

Lior Ben-Gai  
November 2019



## FOR LOOP BLOCK

The diagram illustrates the structure of a for loop block. It consists of three main horizontal segments, each labeled with a red number and text:

- 1. Define Iterator: This segment covers the first part of the initialization section, from the start of the loop to the first semicolon.
- 2. Loop condition: This segment covers the second part of the initialization section, from the first semicolon to the second semicolon.
- 3. Advance Iterator: This segment covers the third part of the initialization section, from the second semicolon to the end of the loop body.

The code itself is written in a stylized font. The first part of the initialization (`var i = 0;`) is colored blue. The loop condition (`i < 100;`) and the iteration advance (`i++`) are both colored red. The loop body, which contains the call to `point()`, is enclosed in black curly braces {} and is colored blue. A vertical yellow bracket on the right side of the body is labeled "Loop body".

```
for(var i = 0; i < 100; i++) {  
    point(i * 10, 10);  
}
```

## WEEK FIVE - ACTION

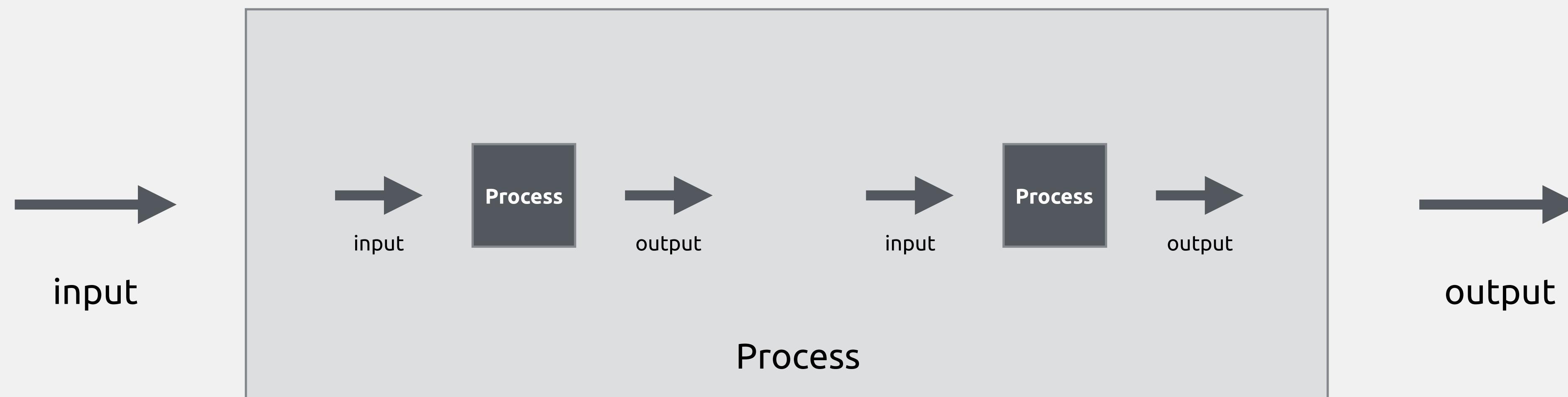
Functions

Lior Ben-Gai  
November 2019

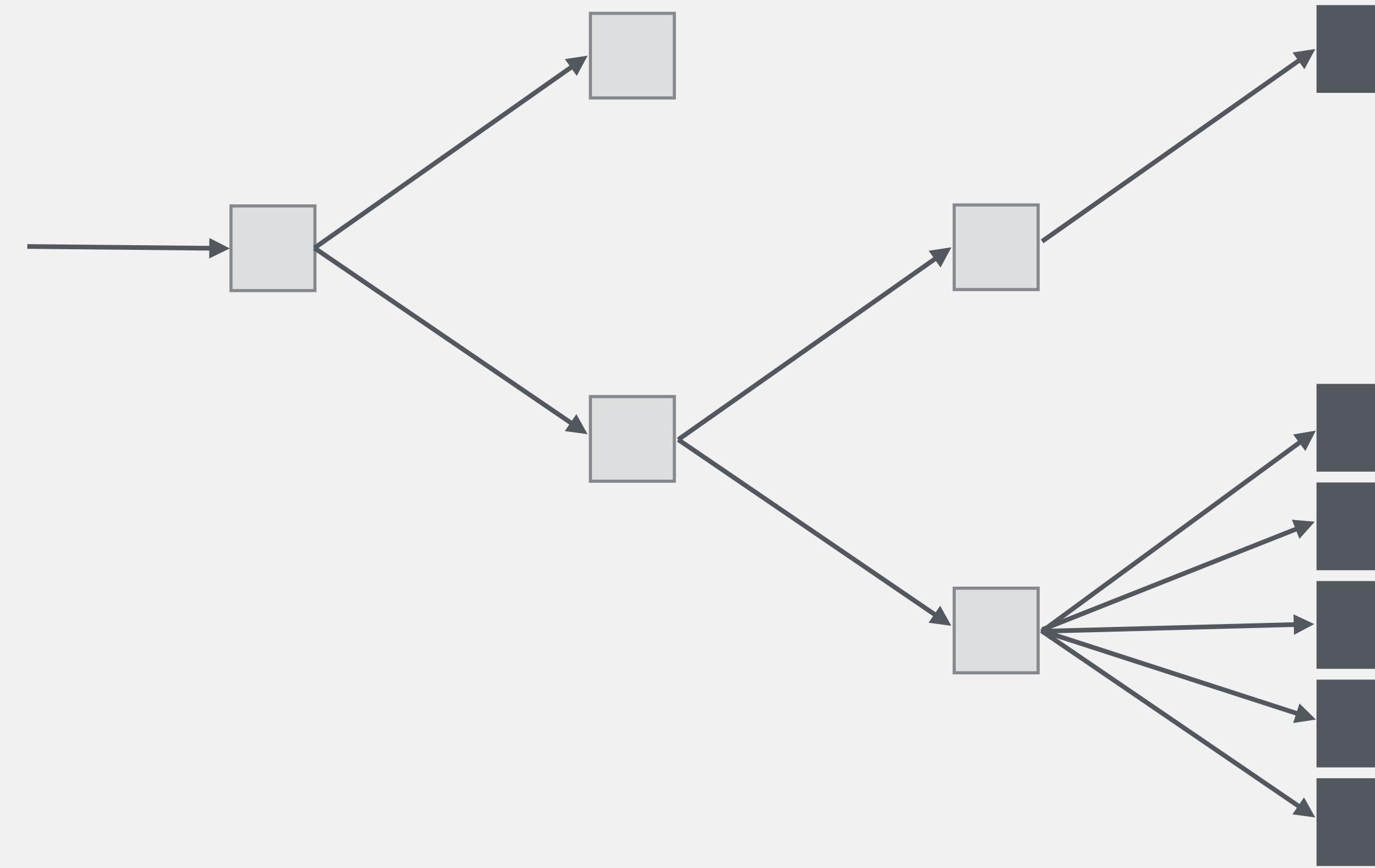
# ENCAPSULATION



# ENCAPSULATION



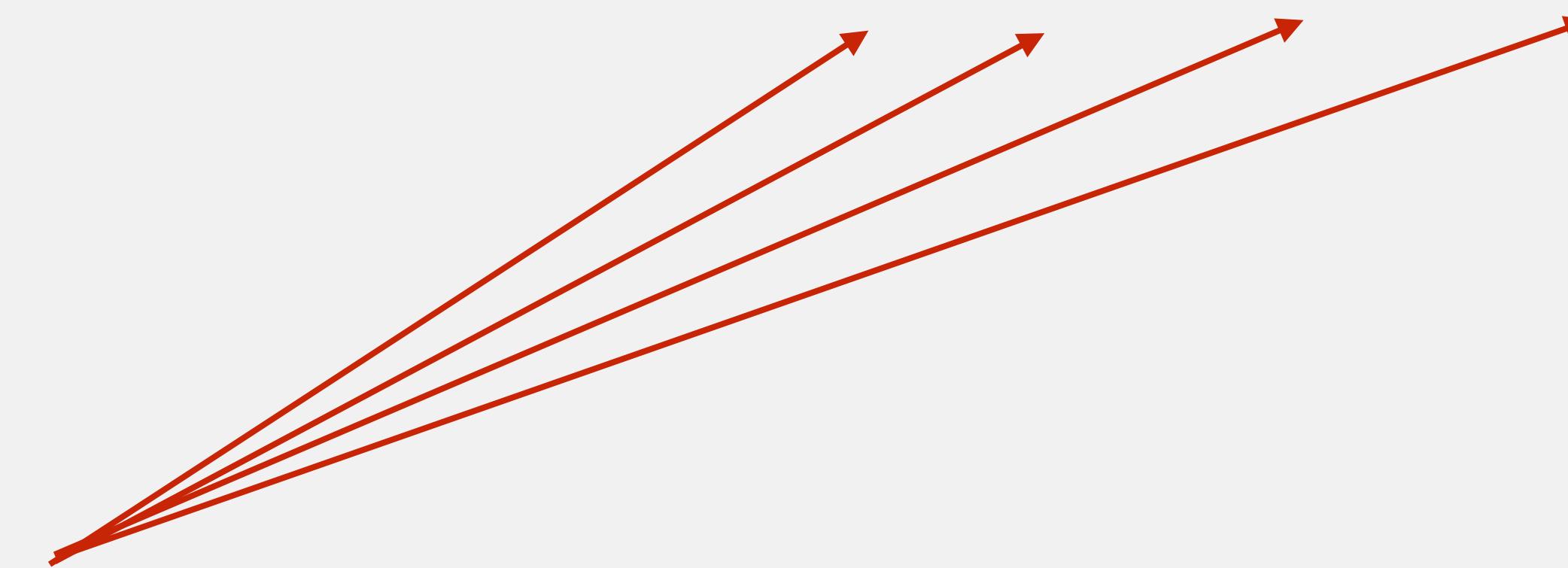
# BRANCHING LOGIC



The combination of conditions, loops and functions allows us to express ANY algorithm

# PARAMETERS

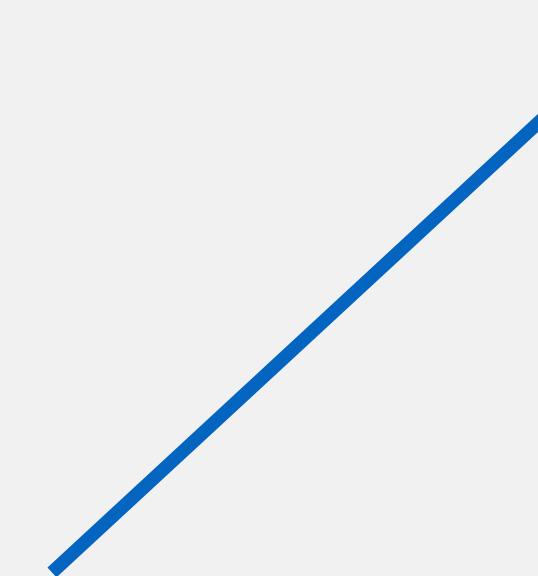
```
line(0, 0, 100, 100);
```



Parameters are bits of information which the function should use  
(input)

## RETURN VALUE

```
var num = random( -10 , 10 );
```



The **Return value** is what replaces the function call inline

(output)

## A FUNCTION DEFINITION

```
sayHello();  
// ...  
function sayHello(){  
    print("Hello!");  
}
```

no Parameters, no Return value

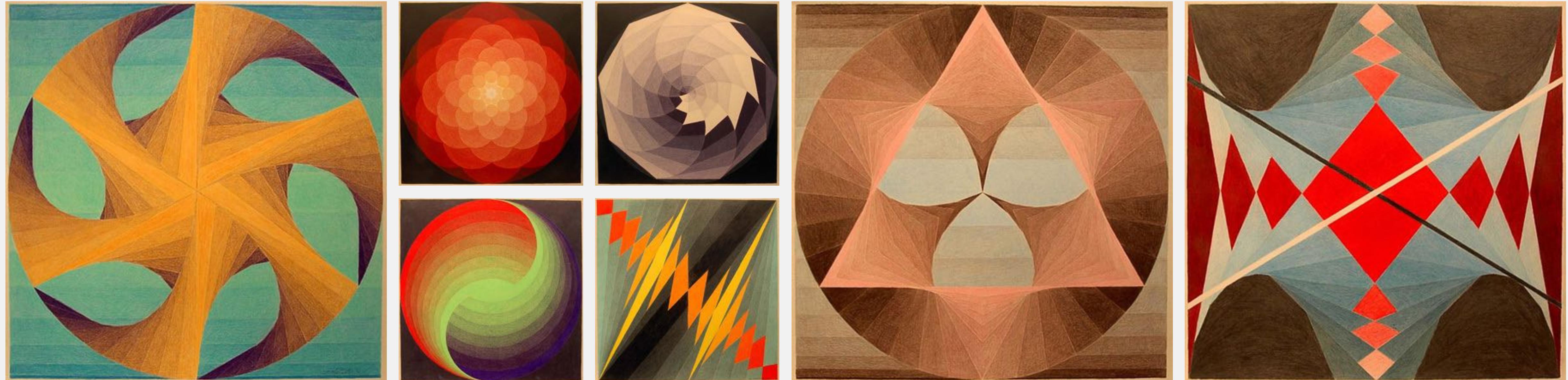
# MODULO

$(40 / 10) \gg 4 \text{ (0)}$  →  $(40 \% 10) \gg 0$

$(42 / 10) \gg 4 \text{ (2)}$  →  $(42 \% 10) \gg 2$



MODULO returns the **remainder** when dividing two numbers



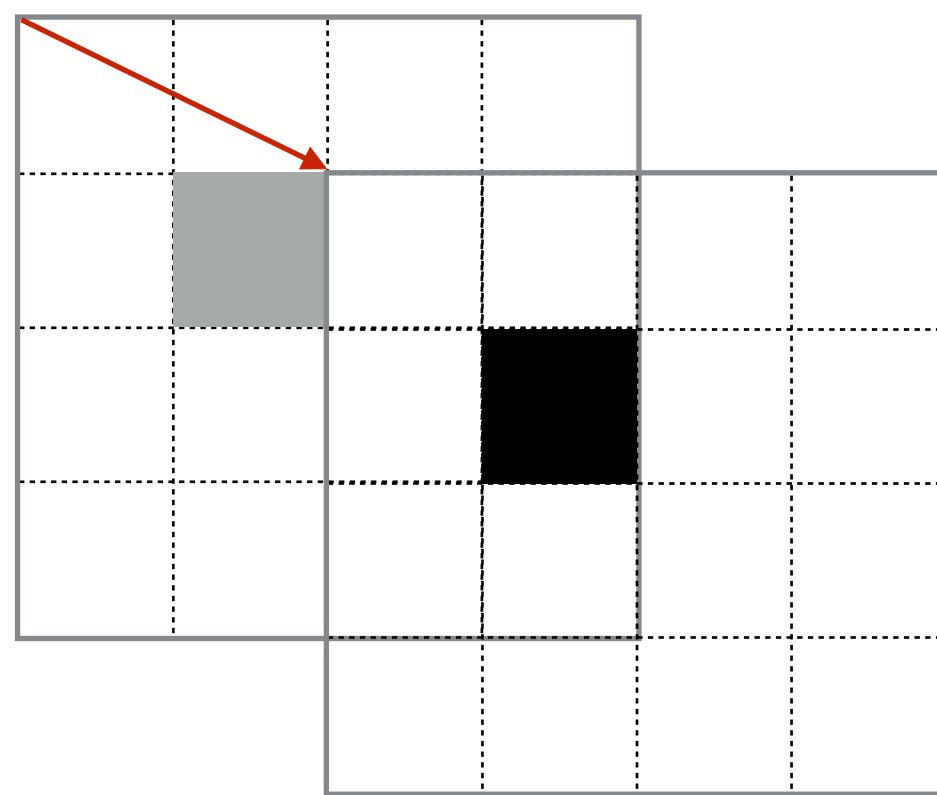
# TRANSFORMATIONS, VARIATIONS

Week 7

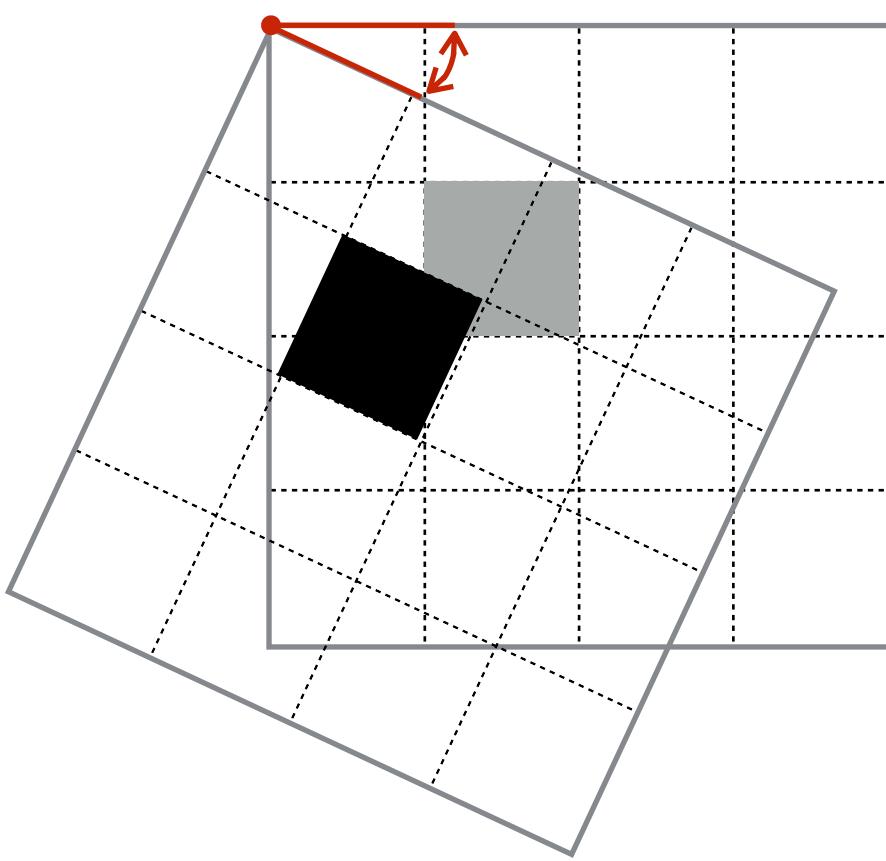
December 2019

Lior Ben-Gai

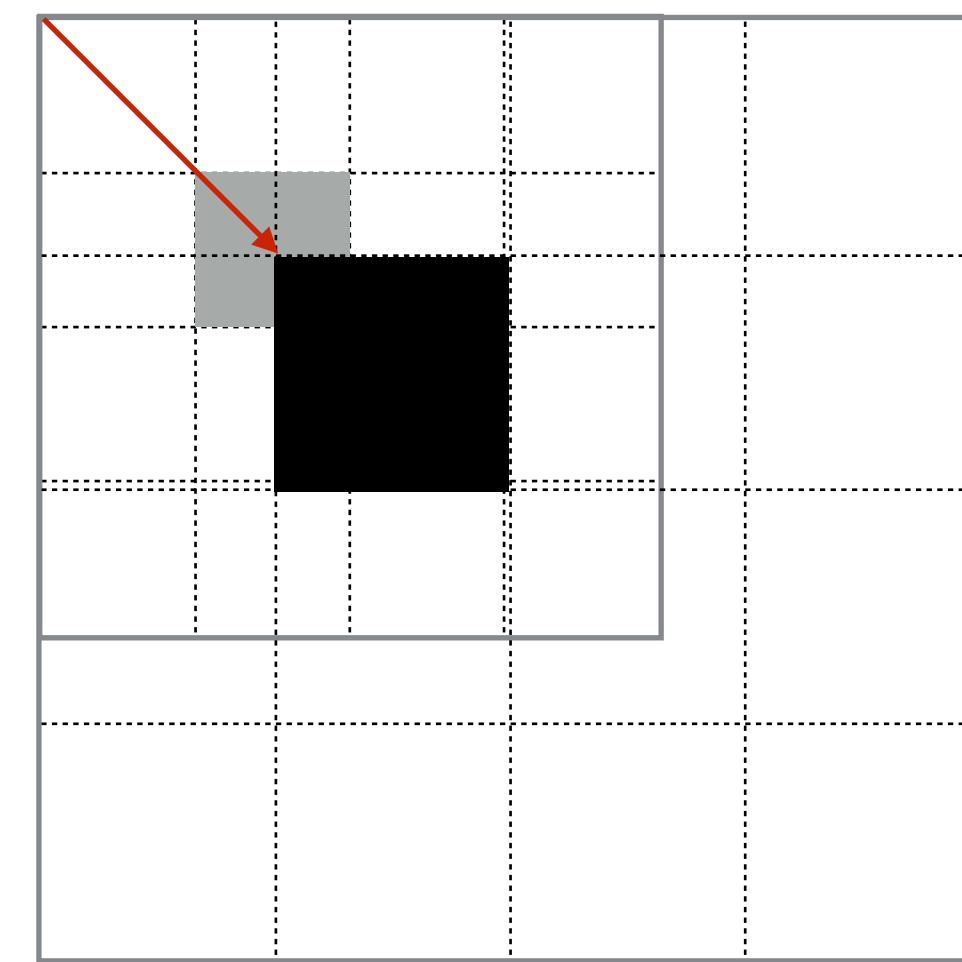
# 2D TRANSFORMATIONS



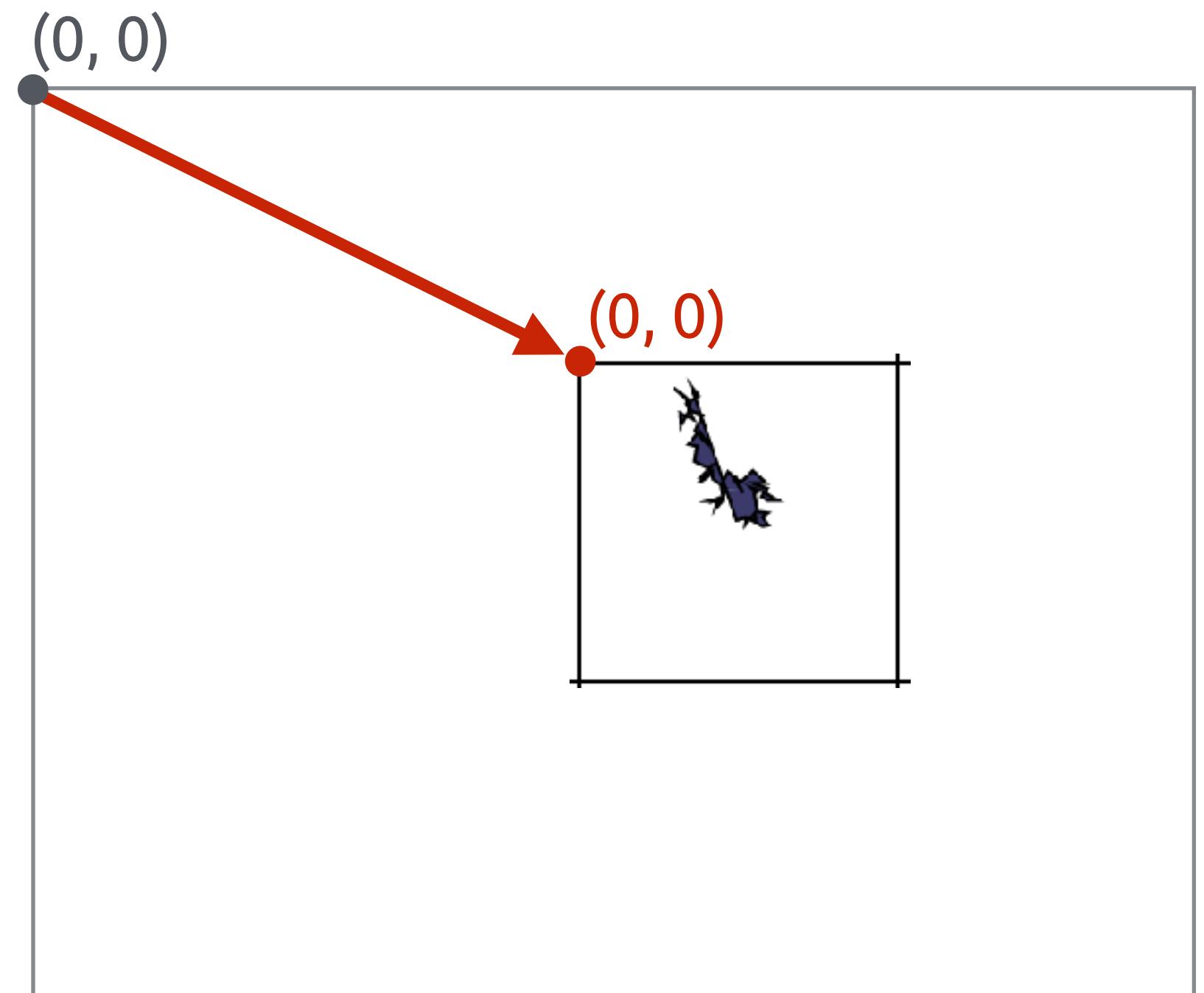
translate



rotate



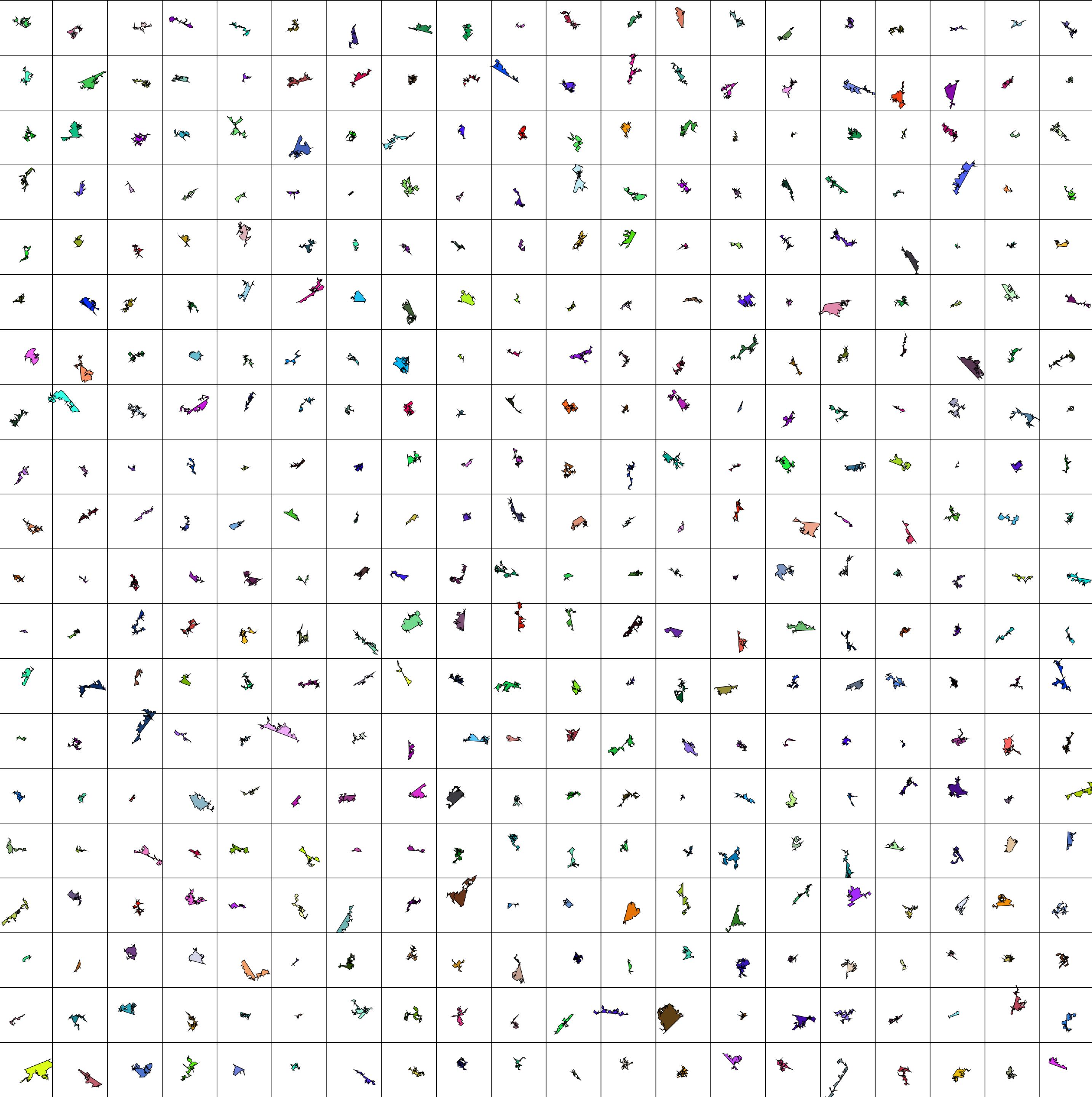
scale



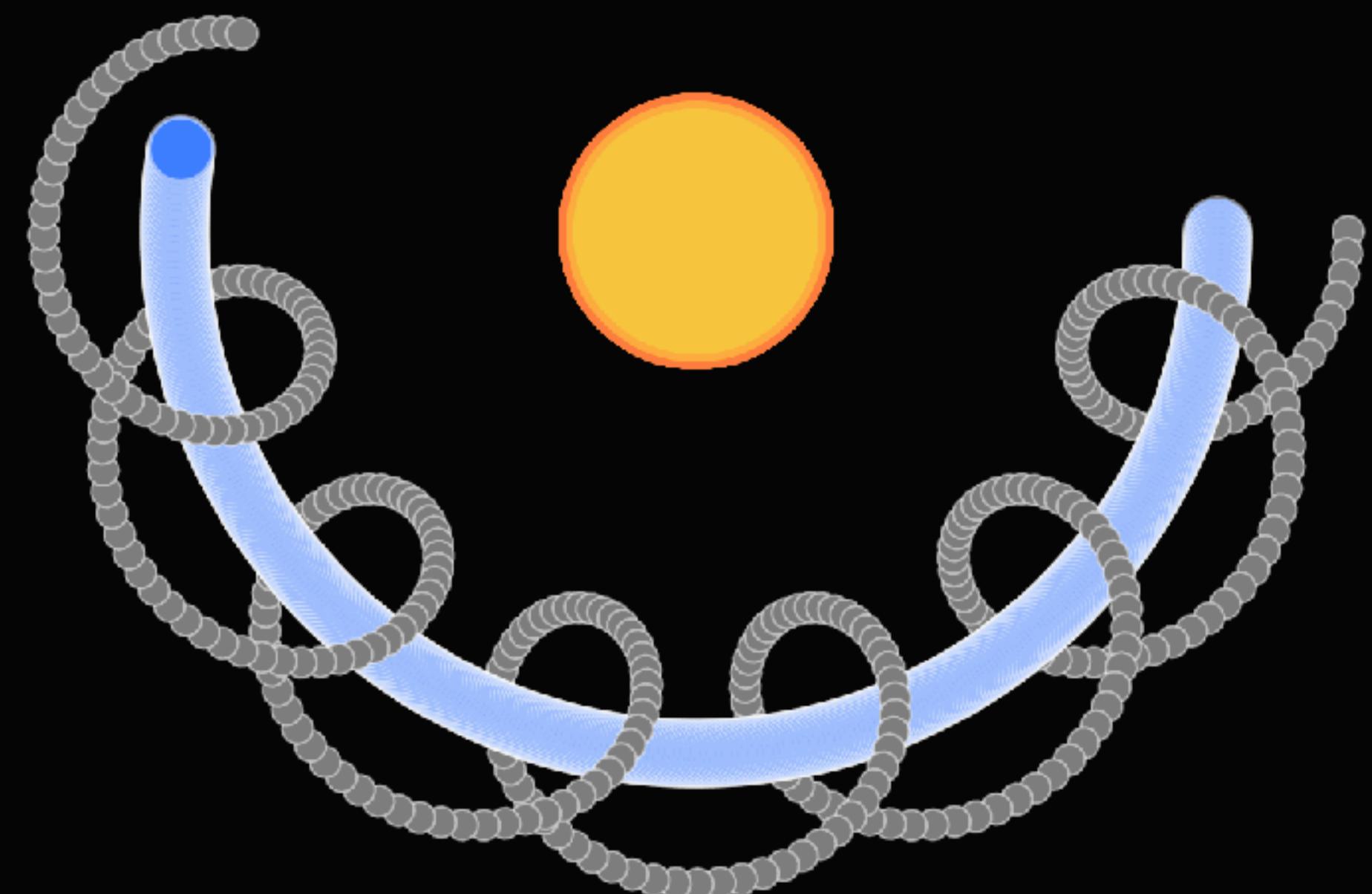
# ORIGAMI

Transformations allow functions to  
draw independently

<https://codepen.io/soogbet/pen/abbeOeY>



compounding orbits

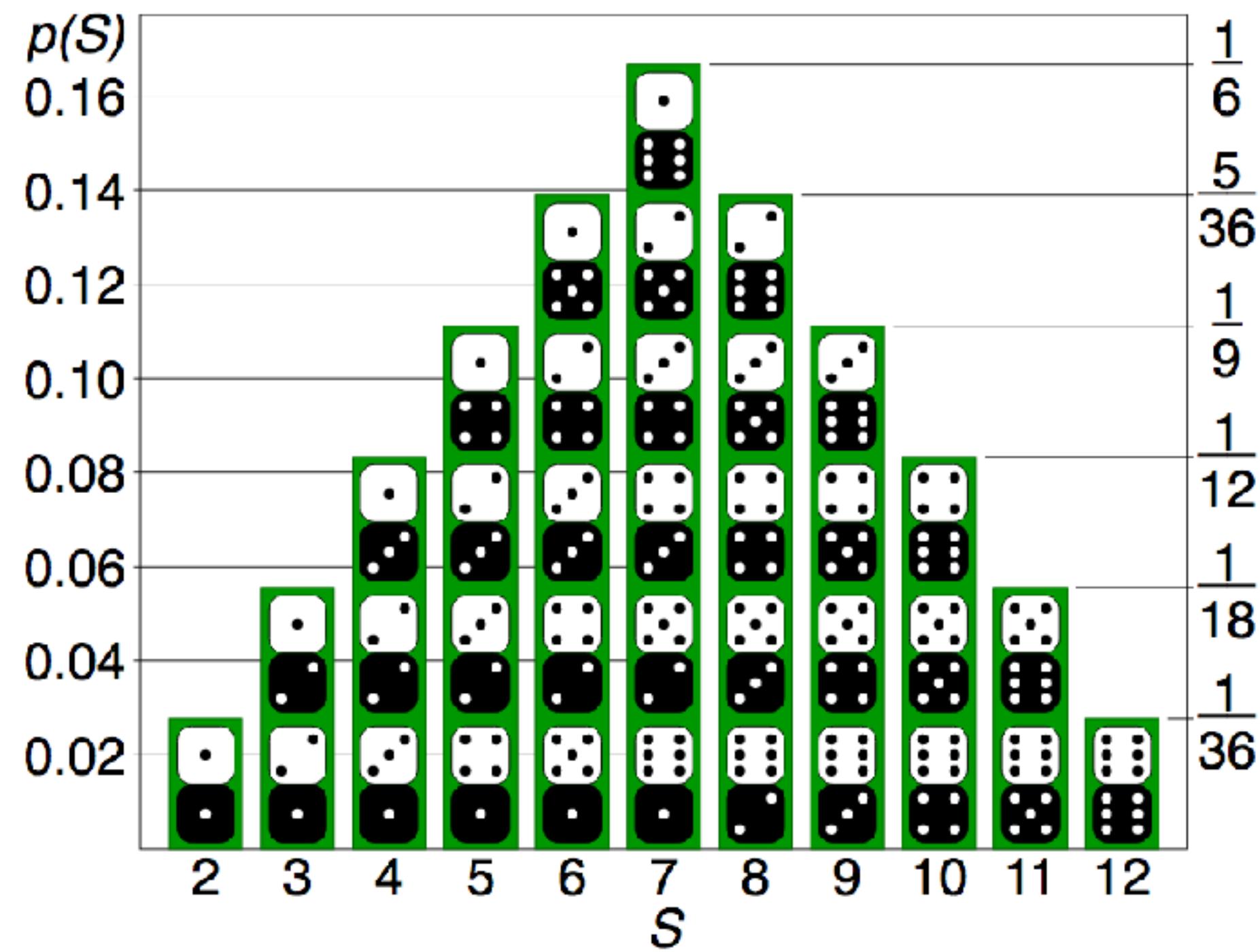
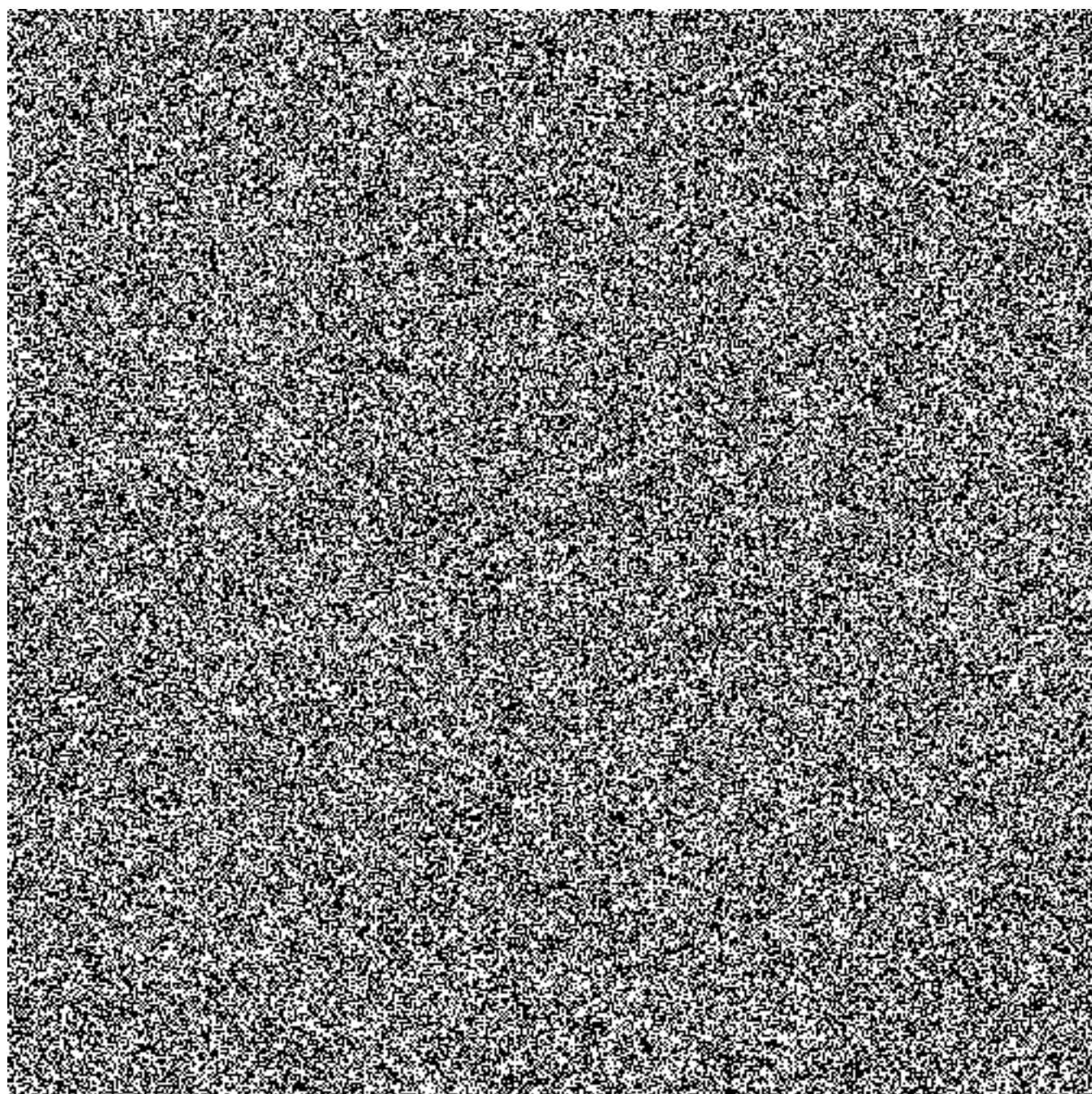


# **GENERATORS**

Procedural content generation techniques

Lior Ben-Gai

December 2019

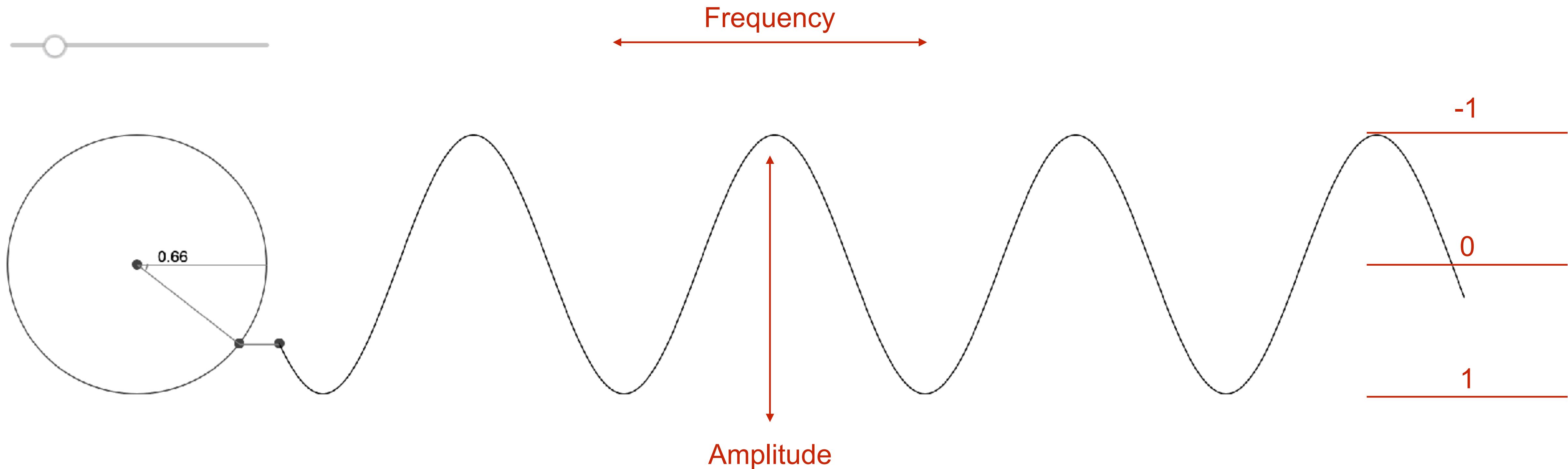


## RANDOMNESS

**Randomness** is the lack of [pattern](#) or [predictability](#) in events.<sup>[1]</sup> A random sequence of events, [symbols](#) or [steps](#) has no [order](#) and does not follow an intelligible pattern or combination. Individual random events are by definition unpredictable, but in many cases the frequency of different outcomes over a large number of events (or "trials") is predictable. For example, when throwing two [dice](#), the outcome of any particular roll is unpredictable, but a sum of 7 will occur twice as often as 4.

<https://en.wikipedia.org/wiki/Randomness>

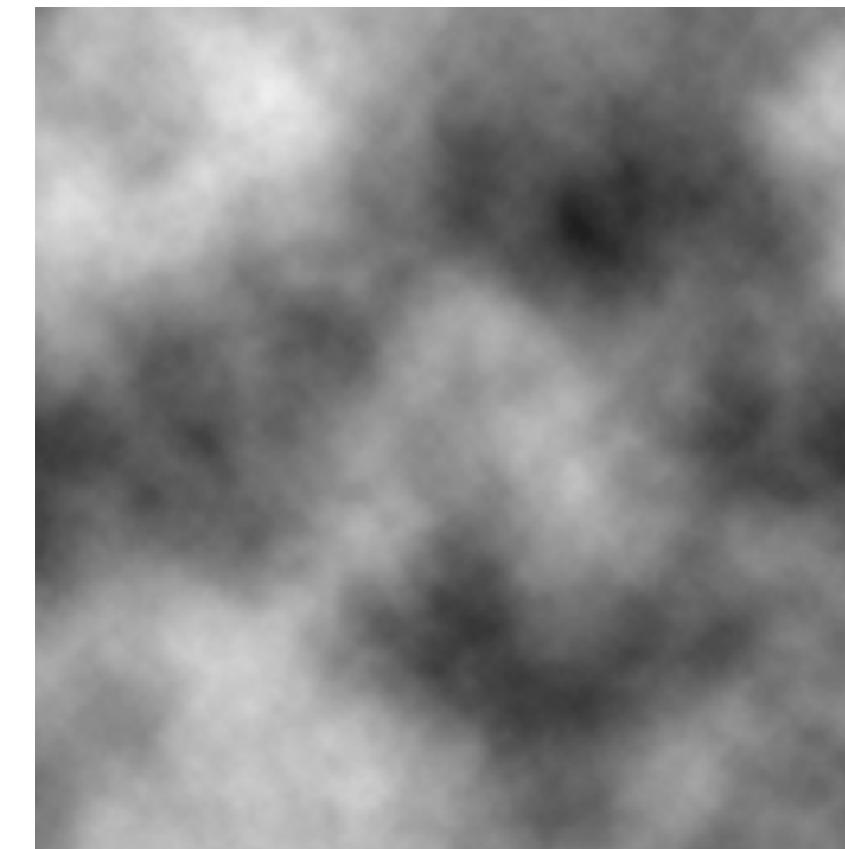
# SINE WAVE



<https://codepen.io/soogbet/pen/eqpLLB?editors=0010>

[https://en.wikipedia.org/wiki/Sine\\_wave](https://en.wikipedia.org/wiki/Sine_wave)

# PERLIN NOISE



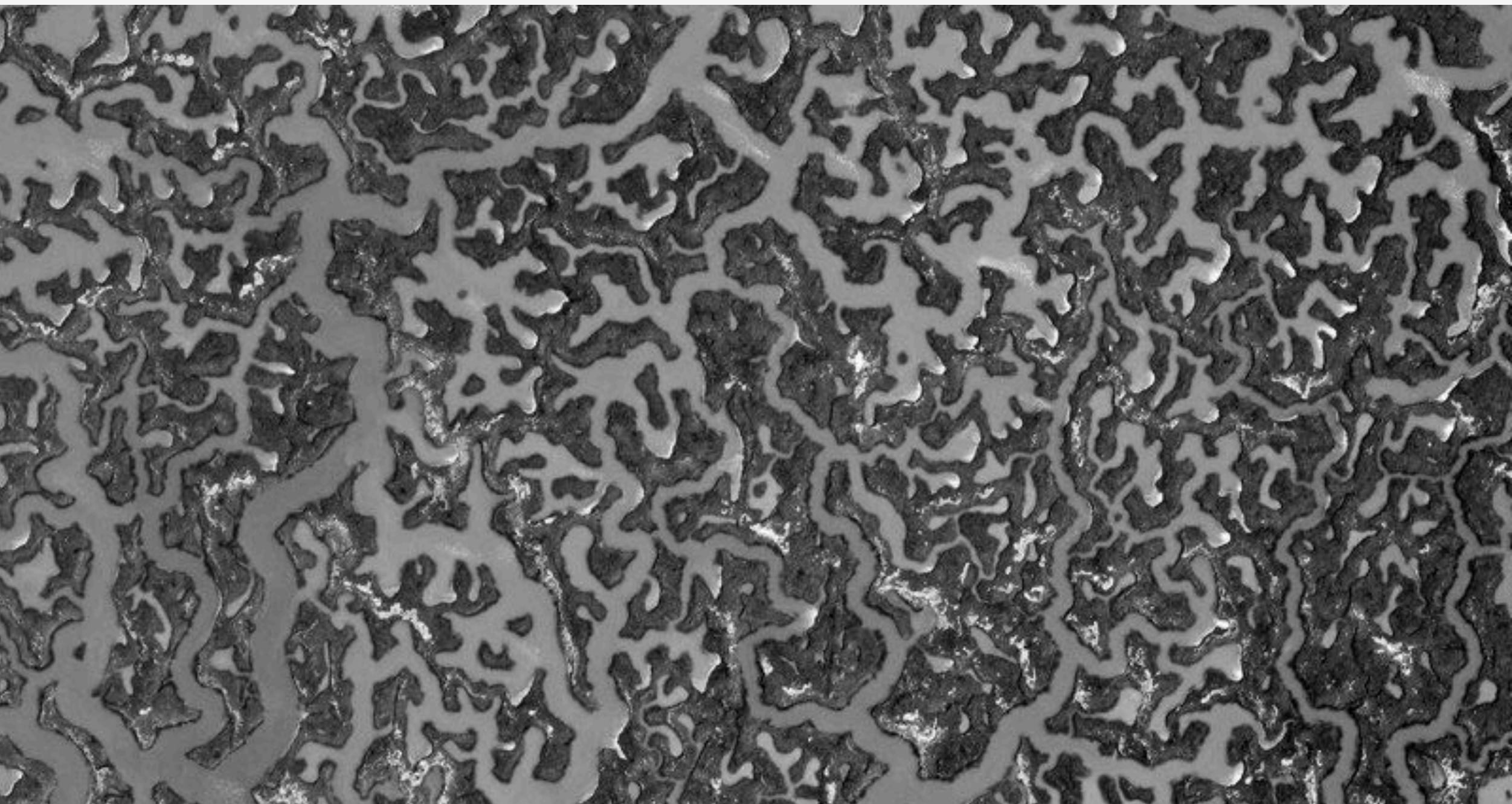
**Perlin noise** is a type of gradient noise developed by [Ken Perlin](#) in 1983 as a result of his frustration with the "machine-like" look of [computer graphics](#) at the time.<sup>[1]</sup> He formally described his findings in a [SIGGRAPH](#) paper in 1985 called *An image Synthesizer*.<sup>[2]</sup> In 1997, Perlin was awarded an [Academy Award for Technical Achievement](#) for discovering the algorithm:<sup>[3]</sup>

[https://en.wikipedia.org/wiki/Perlin\\_noise](https://en.wikipedia.org/wiki/Perlin_noise)

# THE TREE AND THE FOREST

Lior Ben-Gai  
December 2019





BIG



small

# RECURSION

```
‐ for(var i = 0 ; i < 10; i++){  
    doSomething();  
}  
  
‐ function doSomething(){  
    //...  
}
```

```
‐ function doSomething(){  
    doSomethingElse();  
}  
  
‐ function doSomethingElse(){  
    //...  
}
```

```
‐ function doSomething(){  
    doSomething();  
}
```

Iteration

Composition

**Recursion**

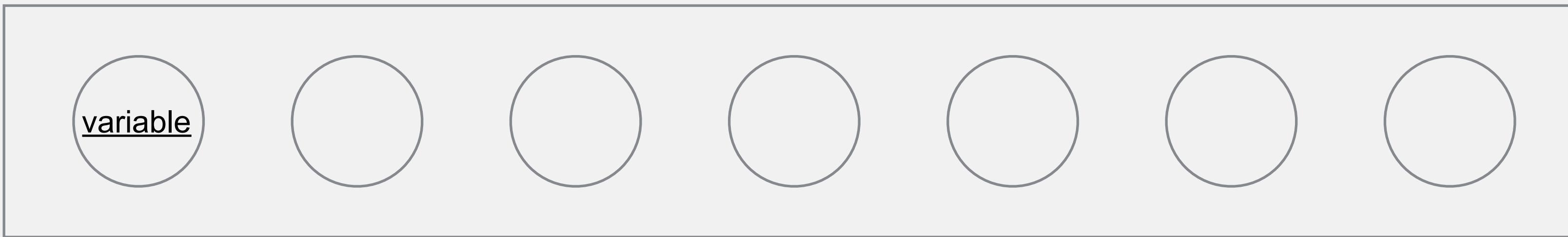
# WEEK TEN - DATA STRUCTURES

**Arrays, Objects, and dynamic data structures**

**Lior Ben-Gai**  
**January 2020**

# DATA STRUCTURES

structure



A **data structure** allows the program to use large amounts of information in order to manage itself.

All data structures are just **lists of variables**. But there are different types of structures, depending on use case and language.

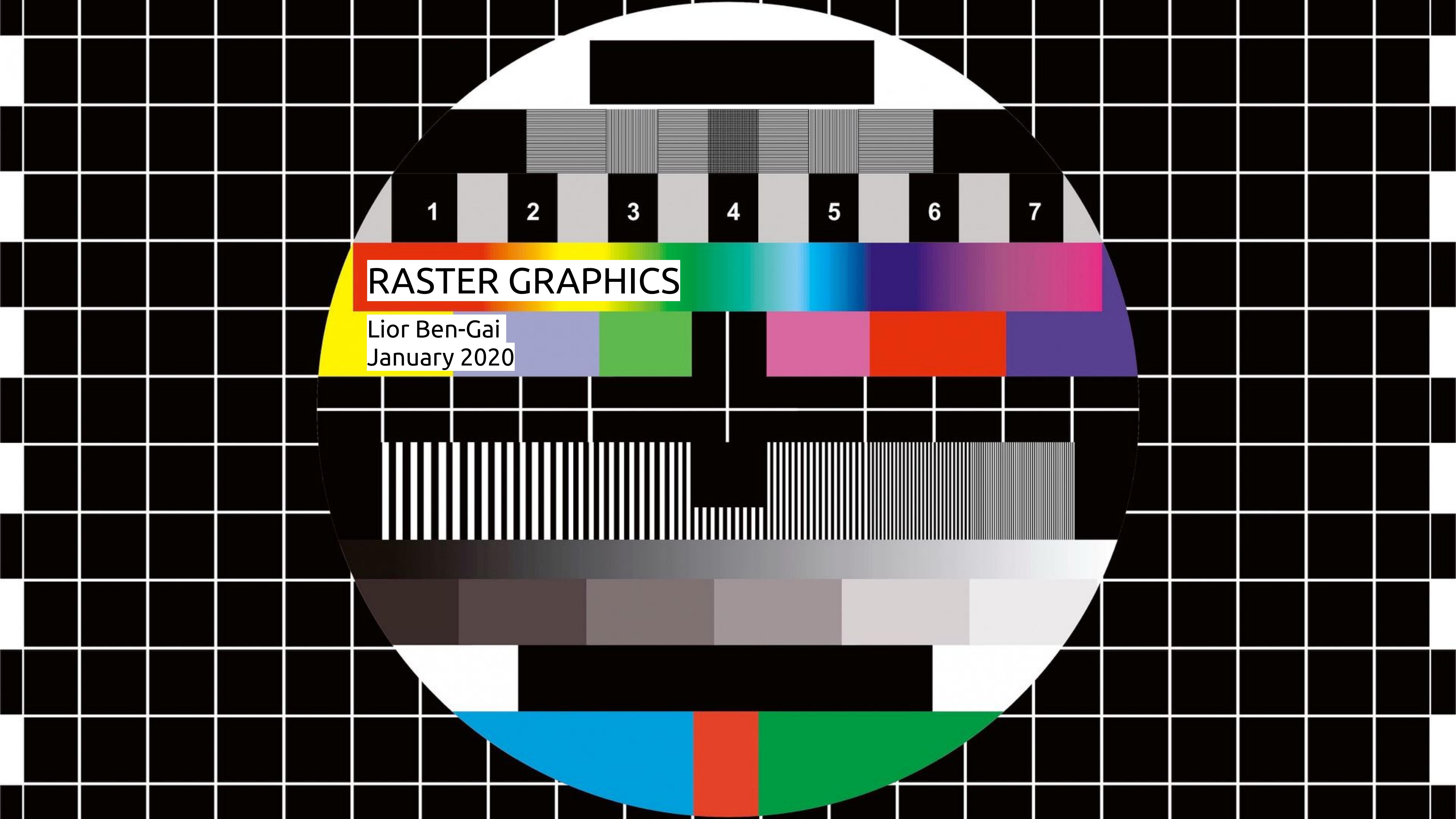
## ARRAYS

```
var positions = [];
```

---

```
for(var i = 0; i < 100; i++) {  
    positions.push( random(width) );  
}
```

**Arrays and loops are best friends!**



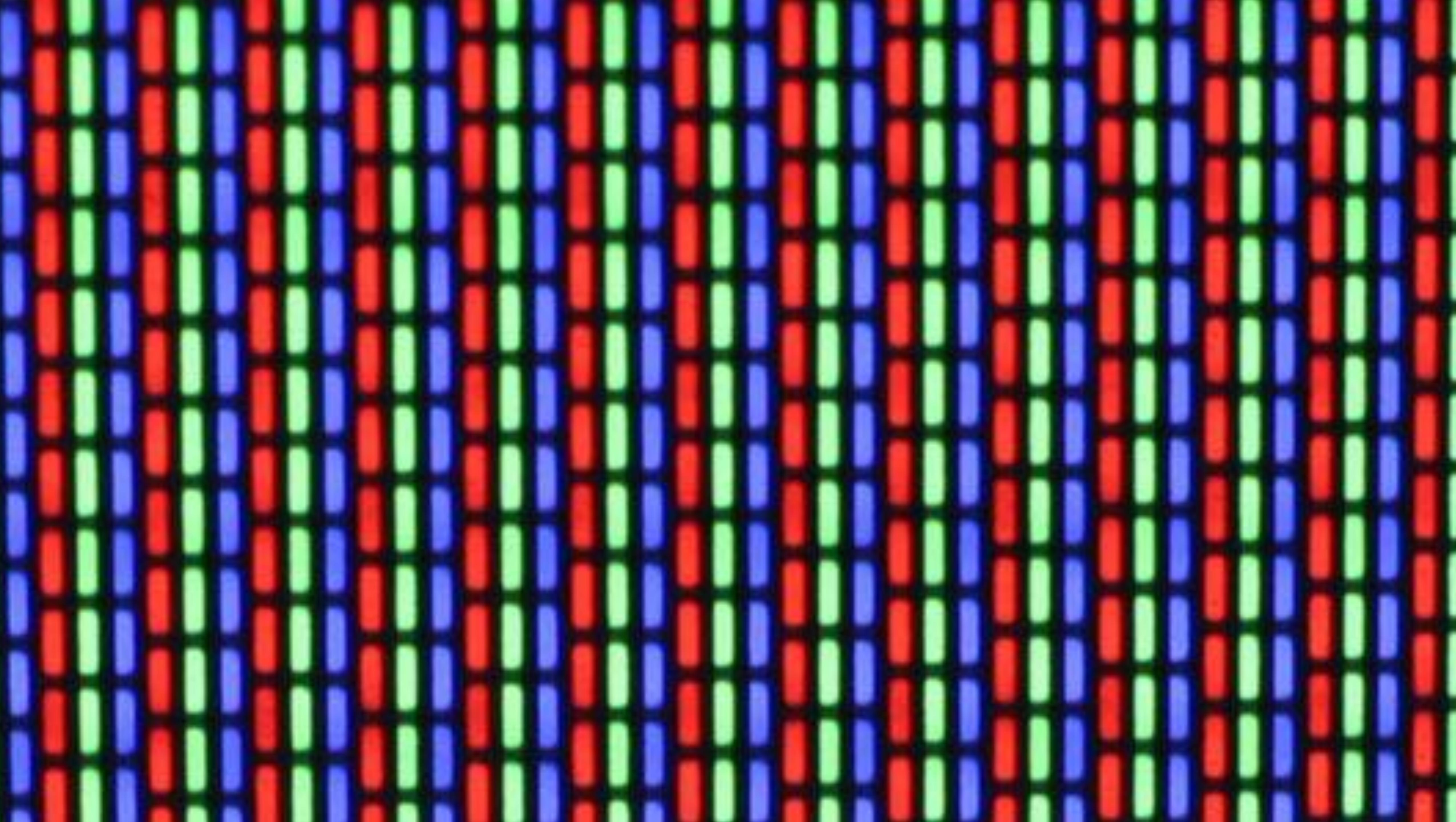
# RASTER GRAPHICS

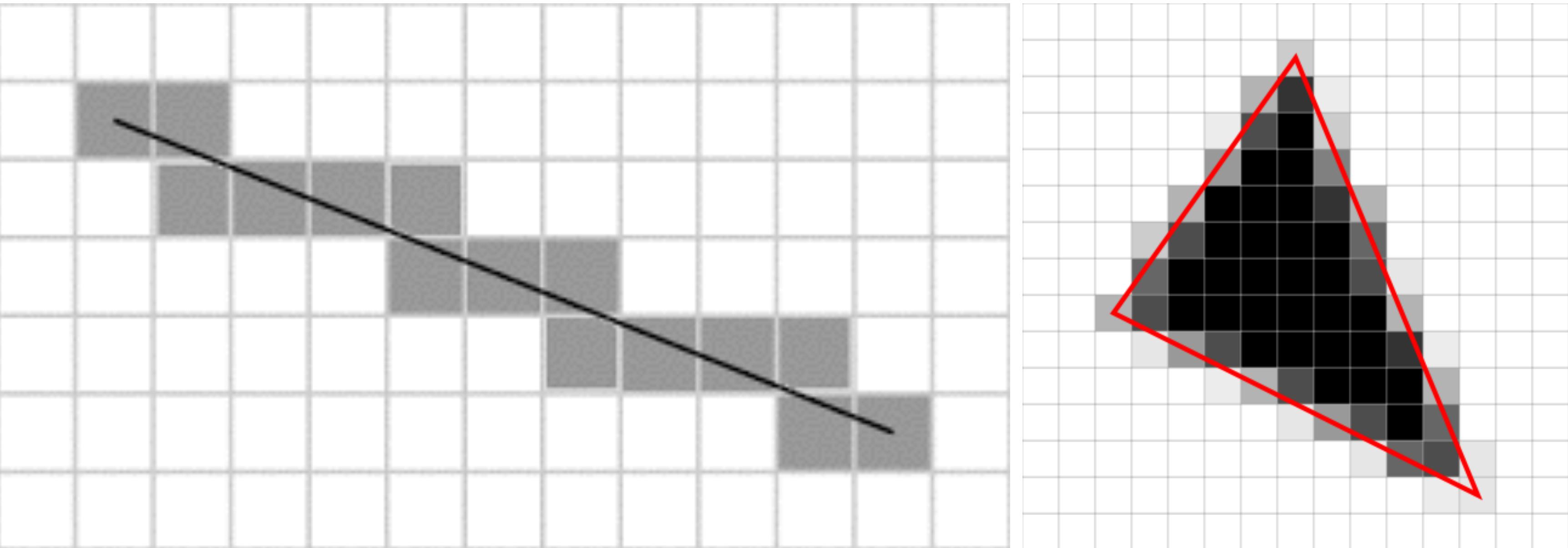
Lior Ben-Gai  
January 2020

-  SoundsOfEarth IS71016B PfA  
h (Lior) (...321 (1))
-  Screen Shot 2017-1...41.35 VOE\_exports
-  Stereo
-  140524\_001.W  
AV.gpk
-  Introduction01
-  UM2\_drawing  
Machi....gcode
-  backpack2.fxg
-  Screen Shot  
2017-1...40.55
-  140524\_001.W  
AV
-  outline\_mk3
-  ImageArchive
-  Macintosh HD



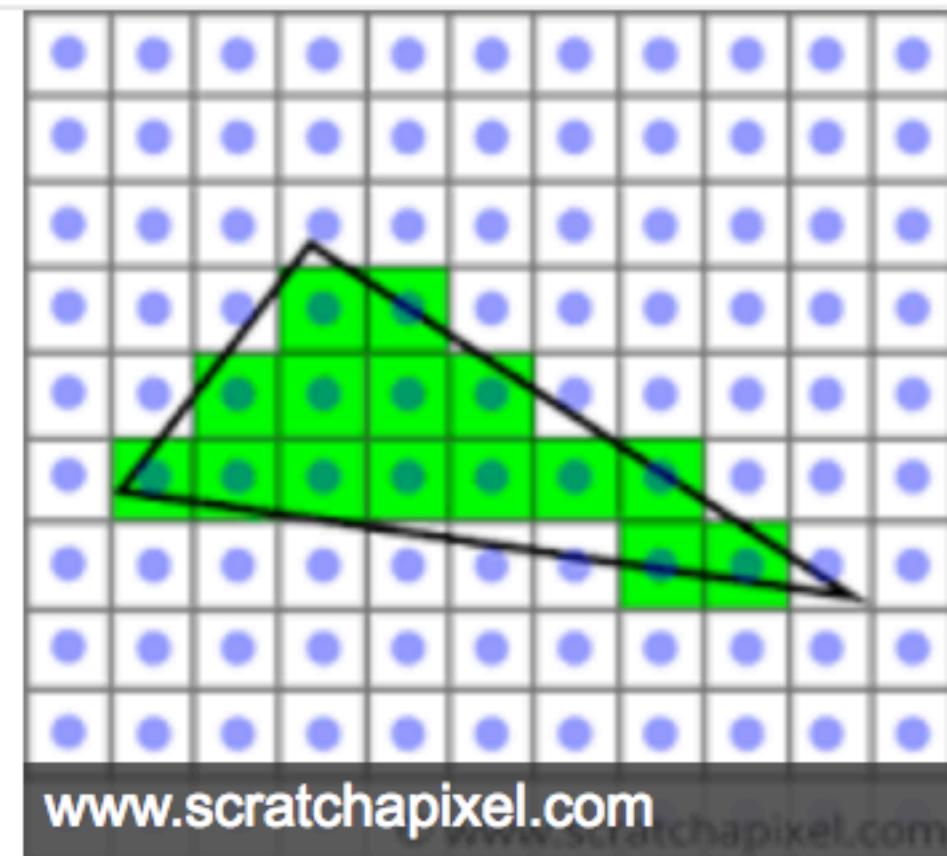






Rasterisation (or **rasterization**) is the task of taking an image described in a vector graphics format (shapes) and converting it into a raster image (pixels or dots) for output on a video display or printer, or for storage in a bitmap file format.

**Rasterisation - Wikipedia**  
<https://en.wikipedia.org/wiki/Rasterisation>



<https://en.wikipedia.org/wiki/Rasterisation>

# INTRO TO WEB DEVELOPMENT

Lior Ben-Gai  
January 2020

## HTML and CSS

Learn HTML  
Learn CSS  
Learn Bootstrap  
Learn W3.CSS  
Learn Colors  
Learn Icons  
Learn Graphics  
Learn How To  
Learn Sass

## JavaScript

Learn JavaScript  
Learn jQuery  
Learn React  
Learn AngularJS  
Learn JSON  
Learn AJAX  
Learn W3.JS

## Server Side

Learn SQL  
Learn PHP  
Learn ASP  
Learn Node.js  
Learn Raspberry Pi

## Programming

Learn Python  
Learn Java  
Learn C++  
Learn C#  
Learn Machine Learning

## Web Building

Web Templates  
Web Statistics  
Web Certificates  
Web Editor  
Web Development

## XML Tutorials

Learn XML  
Learn XML AJAX  
Learn XML DOM  
Learn XML DTD  
Learn XML Schema  
Learn XSLT  
Learn XPath  
Learn XQuery

## References

HTML Reference  
CSS Reference  
JS Reference  
SQL Reference  
PHP Reference  
jQuery Reference  
Python Reference

# HTML

The language for building web pages

LEARN HTML

HTML REFERENCE

## HTML Example:

```
<!DOCTYPE html>
<html>
<title>HTML Tutorial</title>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Try it Yourself »

# CSS

The language for styling web pages

LEARN CSS

CSS REFERENCE

## CSS Example:

```
body {
    background-color: lightblue;
}

h1 {
    color: white;
    text-align: center;
}

p {
    font-family: verdana;
    font-size: 20px;
}
```

Try It Yourself \*

# JavaScript

The language for programming web pages

LEARN JAVASCRIPT

JAVASCRIPT REFERENCE

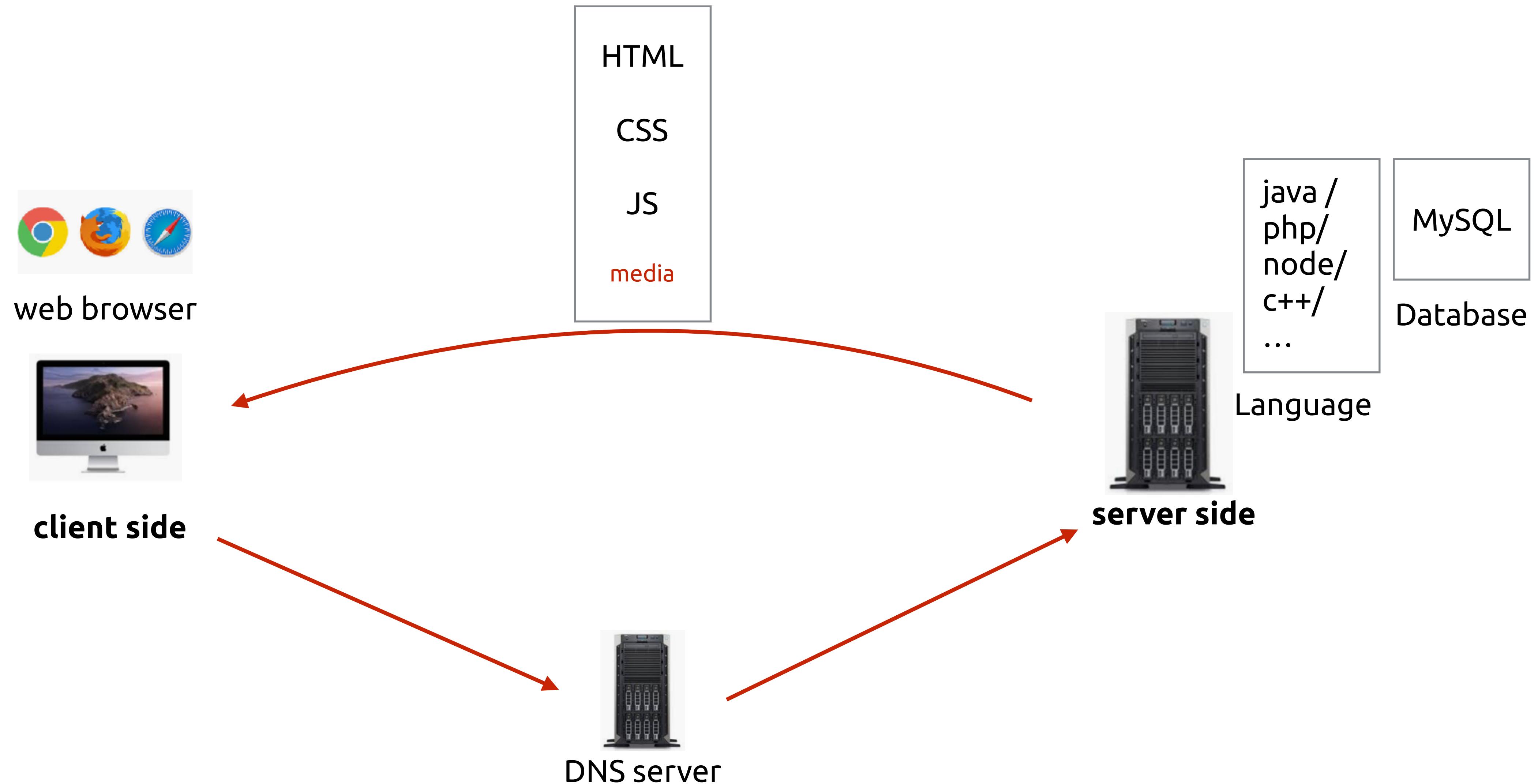
## JavaScript Example:

```
<button onclick="myFunction()">Click Me!
</button>

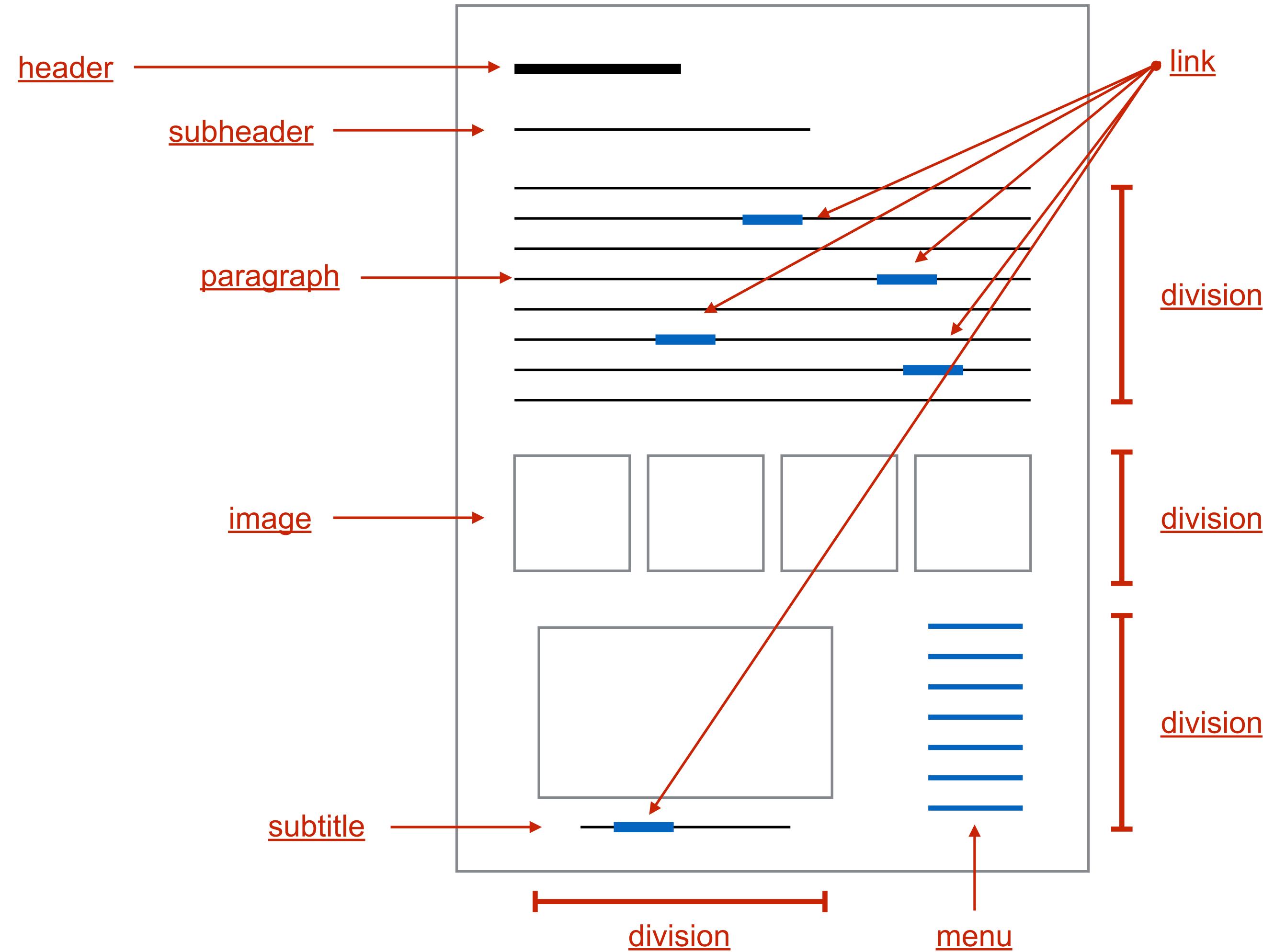
<script>
function myFunction() {
    var x = document.getElementById("demo");
    x.style.fontSize = "25px";
    x.style.color = "red";
}
</script>
```

Try it Yourself »

# WEB TECHNOLOGY



# HYPER TEXT MARKUP LANGUAGE



# HYPER TEXT MARKUP LANGUAGE

We need information **about** the content

in order to determine

how different things **look like**

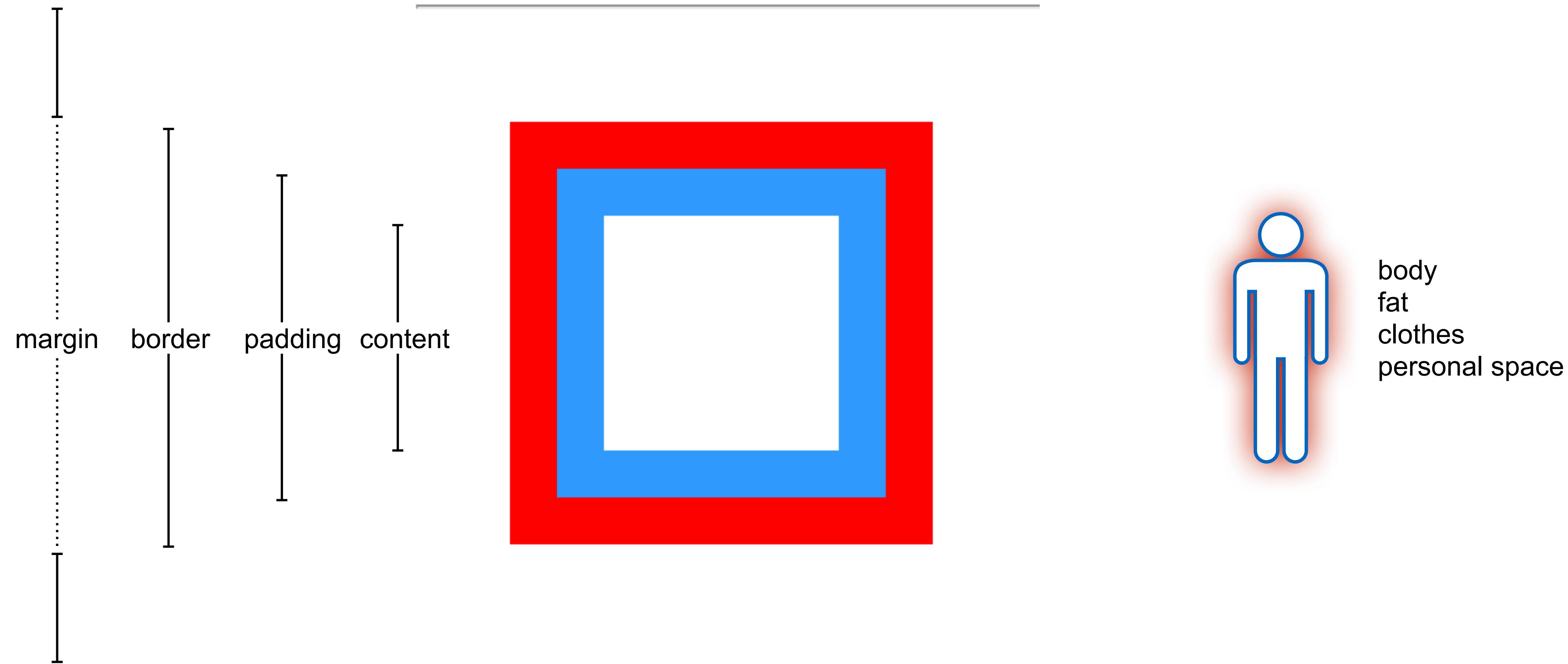
and how they **behave**

HTML

CSS

JS

# CSS BOX MODEL



[https://www.w3schools.com/css/css\\_boxmodel.asp](https://www.w3schools.com/css/css_boxmodel.asp)