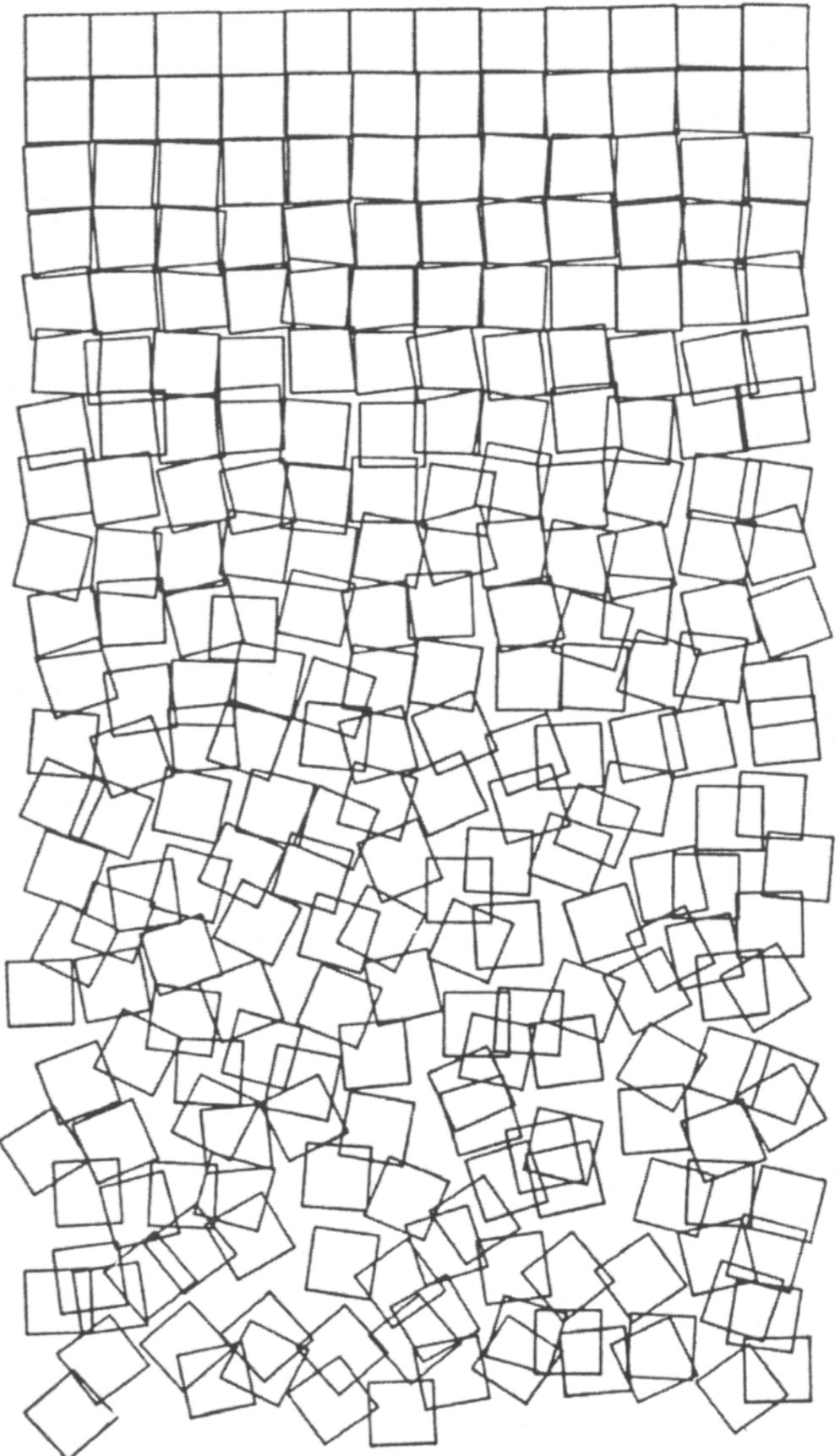


# RANDOMNESS CHANCE AND CREATIVE CODE

DANIEL HOWE  
REDNOISE.ORG/DANIEL  
UNIVERSITY OF THE ARTS  
LONDON, JULY 9, 2021



*Schotter (Gravel)* - Georg Nees, 1968

- THEORETICAL: DEFINITIONS
  - AESTHETIC: EXAMPLES IN ART
  - CONCEPTUAL: WHY USE RANDOMNESS?
  - TECHNICAL: USING RANDOMNESS EFFECTIVELY
- 

# OBJECTIVES

# **RESOURCES**

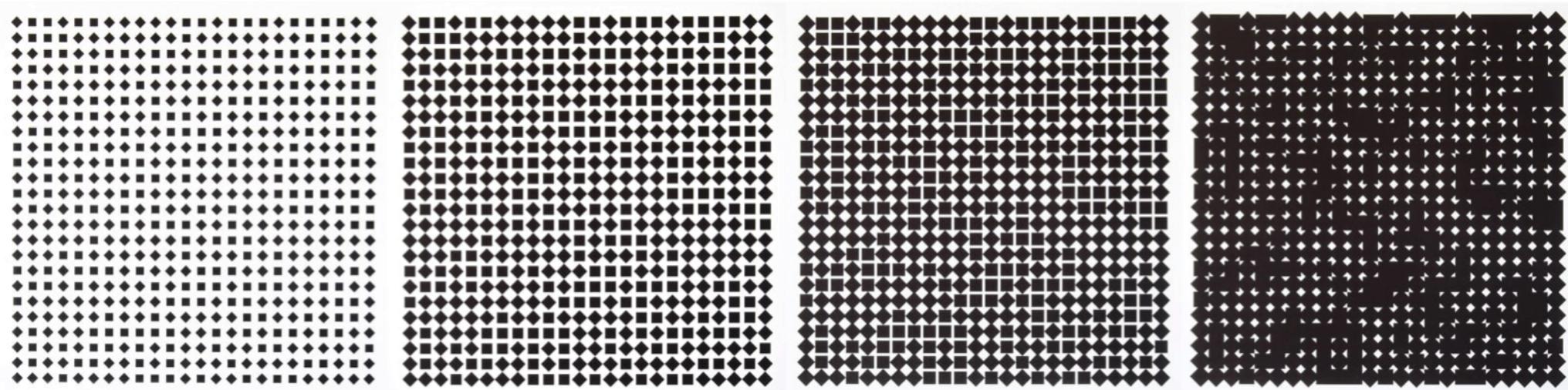
---

**TOOLS & LINKS**

# SLIDES + CODES



<https://github.com/dhowe/rws>



VERA MOLNAR, CARRÉS EN  
2 POSITIONS 1-4 , 2011-13

[dhowe / GetGen](#)

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

A Workshop for Processing Community Day @ HK 2019 Edit

Manage topics

19 commits 1 branch 0 releases 1 contributor

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

dhowe updated Latest commit 49854aa 27 seconds ago

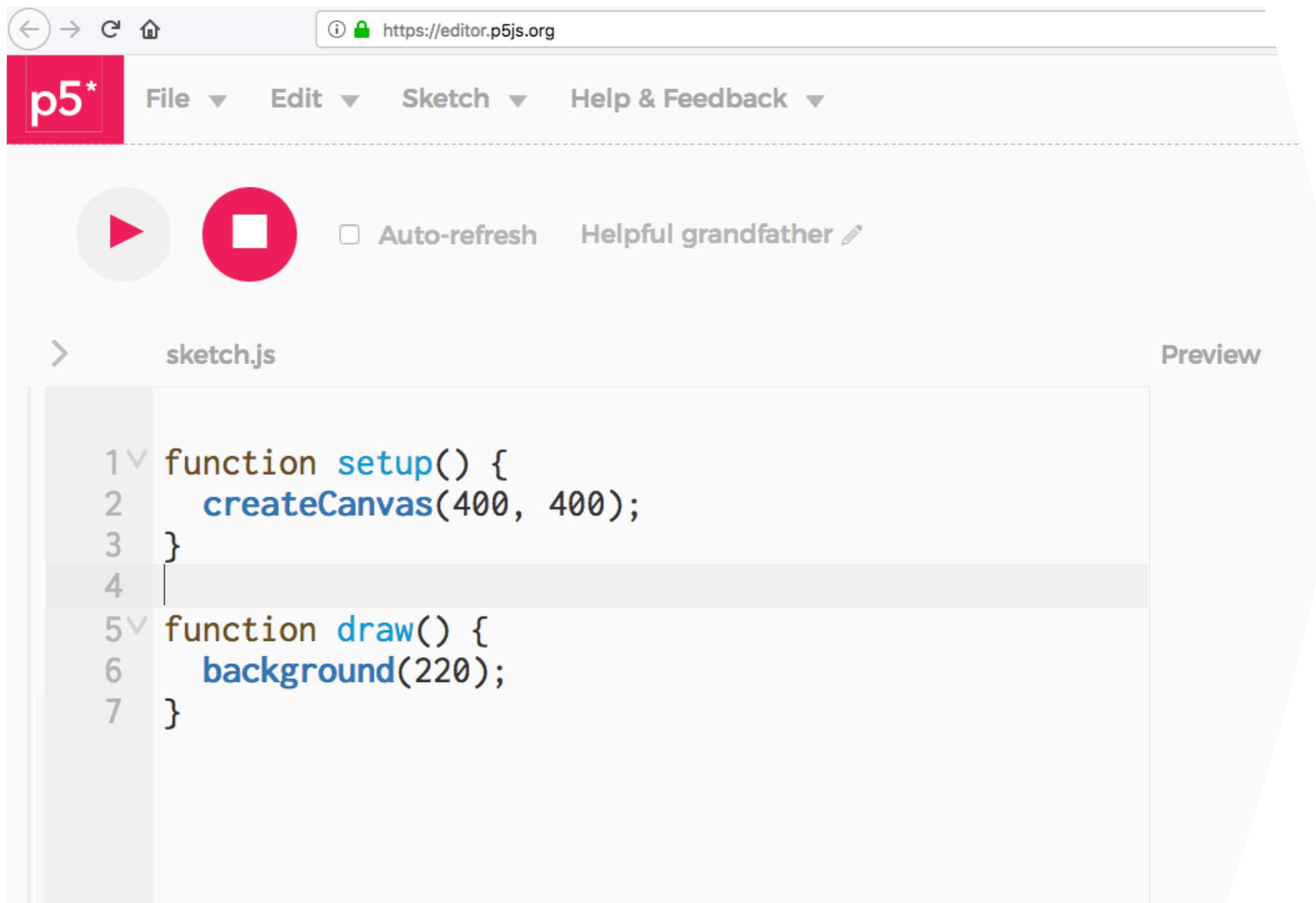
File	Message	Time
README.md	Update README.md	23 hours ago
basic-tree.js	initial	a day ago
fractal-tree1.js	initial	a day ago
fractal-tree2.js	initial	a day ago
fractal-tree3.js	initial	a day ago
getgen.png	initial	23 hours ago
mapped-tree.js	initial	a day ago
slides.pdf	updated	26 seconds ago

README.md

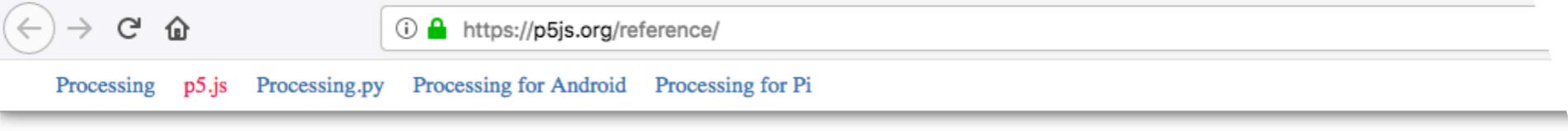


<https://github.com/dhowe/GetGen>

# TOOLS: P5.JS EDITOR



# TOOLS: REFERENCE



The screenshot shows the header of the p5.js reference site. It includes browser navigation icons (back, forward, search, home), a URL bar with <https://p5js.org/reference/>, and a navigation menu with links to Processing, p5.js, Processing.py, Processing for Android, and Processing for Pi.

**p5.js**  
Processing creativity times JavaScript dynamism

**Reference**

**Home**

**Download** Can't find what you're looking for? You may want to check out [here](#).

**Start** You can download an offline version of the reference [here](#).

**Reference**

<a href="#">Color</a>	<a href="#">Environment</a>	<a href="#">Lights, Camer</a>
<a href="#">Constants</a>	<a href="#">Events</a>	<a href="#">Math</a>
<a href="#">DOM</a>	<a href="#">IO</a>	<a href="#">Rendering</a>
<a href="#">Data</a>	<a href="#">Image</a>	<a href="#">Shape</a>

**Examples**

**Books**

**Community**

**Color**

<a href="#">Creating &amp;</a>	<a href="#">Setting</a>
<a href="#">Reading</a>	<a href="#">background()</a>

**Search**

# THEORETICAL

---

CAN WE DEFINE RANDOMNESS ?

**HOW TO DEFINE RANDOMNESS ?**

---

**RANDOMNESS**

**IS '2' A RANDOM NUMBER ?**

---

**RANDOMNESS**

HOW DOES RANDOMNESS  
RELATE TO PREDICTABILITY?

---

RANDOMNESS

A NUMBER IS RANDOM WHEN THERE IS AN EQUAL CHANCE FOR IT TO BE SELECTED FROM A SET OF POSSIBLE VALUES...

---

# RANDOMNESS

**REDUCIBILITY:** CAN A SEQUENCE OF  
NUMBERS BE EXPRESSED MORE CONCISELY  
THAN BY SIMPLY LISTING THE SEQUENCE?

---

# RANDOMNESS

from Kolmogorov, 1965



**CONSIDER THE FOLLOWING TWO  
SEQUENCES OF 20 COIN FLIPS:**

- A. HTHHTTTHTTTHTHHTHTHH
- B. TTTTTTTTTTTTTTTTTTT

**WHICH IS MORE LIKELY, A OR B?**

---



ACCORDING TO PROBABILITY, THE TWO ARE EQUIALLY LIKELY, EACH HAVING A CHANCE OF 1 IN 1024

- A. HTHHTTTHTTTTHHTHHHTHTHH
- B. TTTTTTTTTTTTTTTTTTT

BUT WHICH SEQUENCE IS MORE RANDOM?

---



ACCORDING TO PROBABILITY, THE TWO ARE  
EQUALLY LIKELY, EACH HAVING A CHANCE OF  
1 IN ABOUT A MILLION...

- A. HTHHTTTHTTTHTHHHTHTH
- B. TTTTTTTTTTTTTTTTTTT

BUT WHICH SEQUENCE IS MORE RANDOM?

---

LET'S DESCRIBE EACH AS CONCISELY AS POSSIBLE

B. TTTTTTTTTTTTTTTTTTT

→ *do T 20 times*

A. HTHTHTHTHTHTHHHTHTH

→ ???

FOR A, THE BEST WE CAN DO IS TO LIST THE WHOLE SEQUENCE ITSELF...

*HTHTHTHTHTHTHHHTHTH*



# LET'S ADD ONE MORE SEQUENCE...

C. TTFFFTTFFTFFTTFFTTFF

???

B. TTTTTTTTTTTTTTTTT

*do T 20 times*

A. HTHHTTTHTTTHTHHHTHTH

*do HTHHTTTHTTTHTHHHTHTH*



LESS  
RANDOM  
(SHORTER)



B. TTTTTTTTTTTTTTTTT

*do T 20 times*

C. TTFFTTFFTTFFTTFF

*do TTFF 5 times*

A. HTHTTHTHTTHHHTHTH

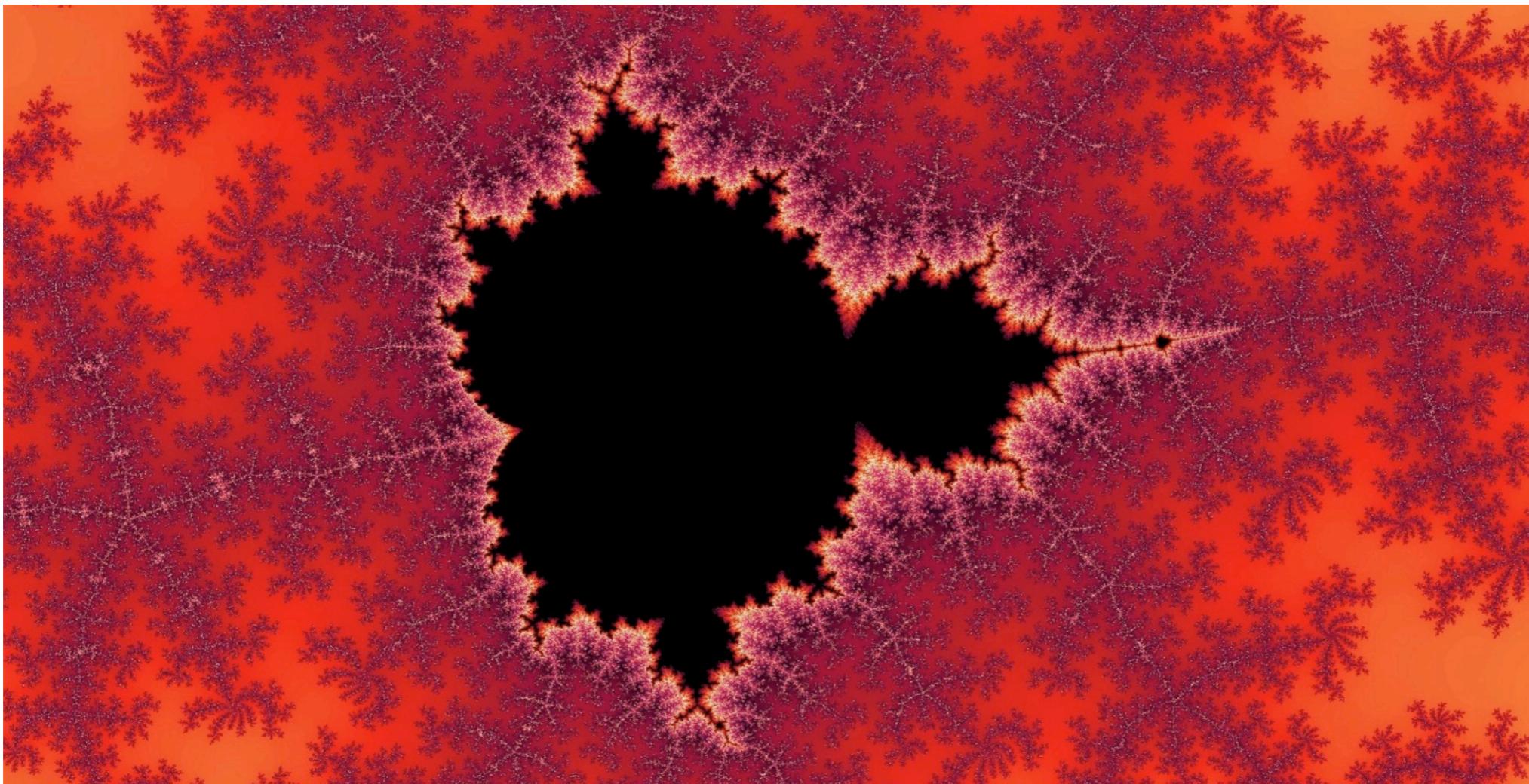
*do HTHTHTTHTHTHHTHTH*

MORE  
RANDOM  
(LONGER)

from Kolmogorov, 1965

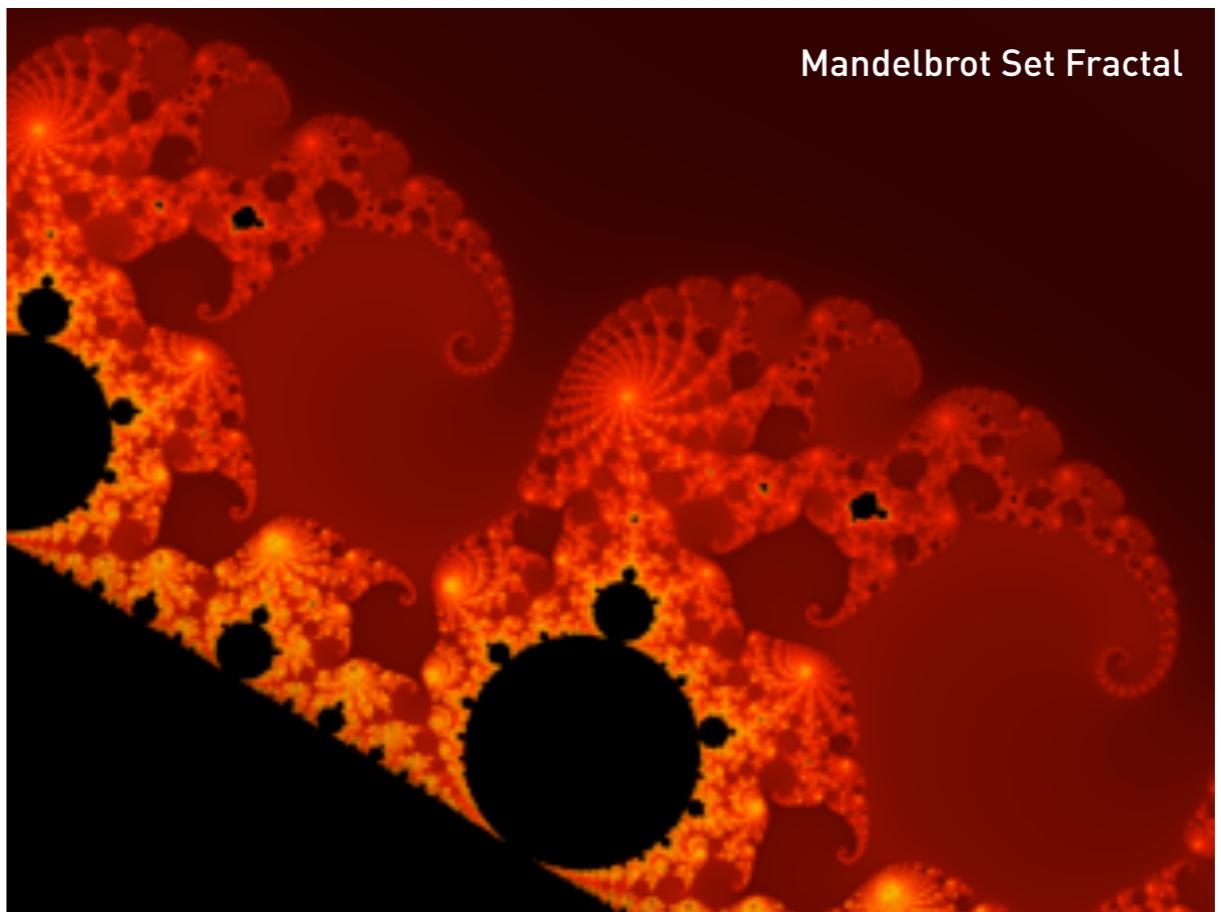
THE THEORY OF KOLMOGOROV COMPLEXITY  
IS BASED ON A SIMPLE IDEA:

*Complex objects cannot be  
described with a short program*



# KOLMOGOROV COMPLEXITY

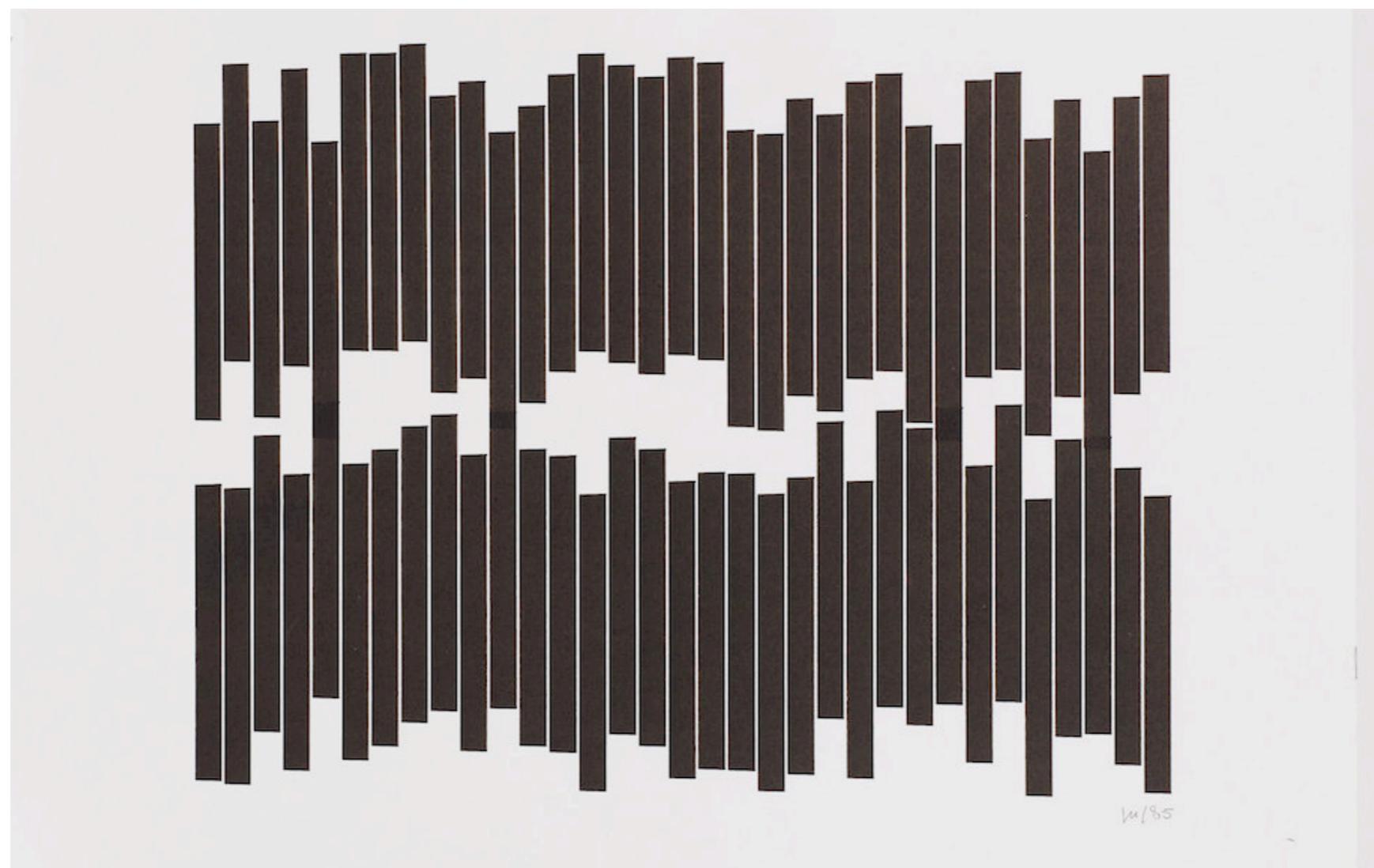
- Storing the 24-bit color of each pixel in this Mandelbrot set fractal would require around 1.6 million bytes
- Yet, a short program can reproduce the image using the basic definition of the Mandelbrot set
- Thus, the complexity of the image is far less than 1.6 million bytes



KOLMOGOROV'S APPROACH WAS ANOTHER  
WAY OF ASKING ...

IS THERE A PATTERN TO THE SEQUENCE?

---



# RANDOMNESS IN ART + DESIGN

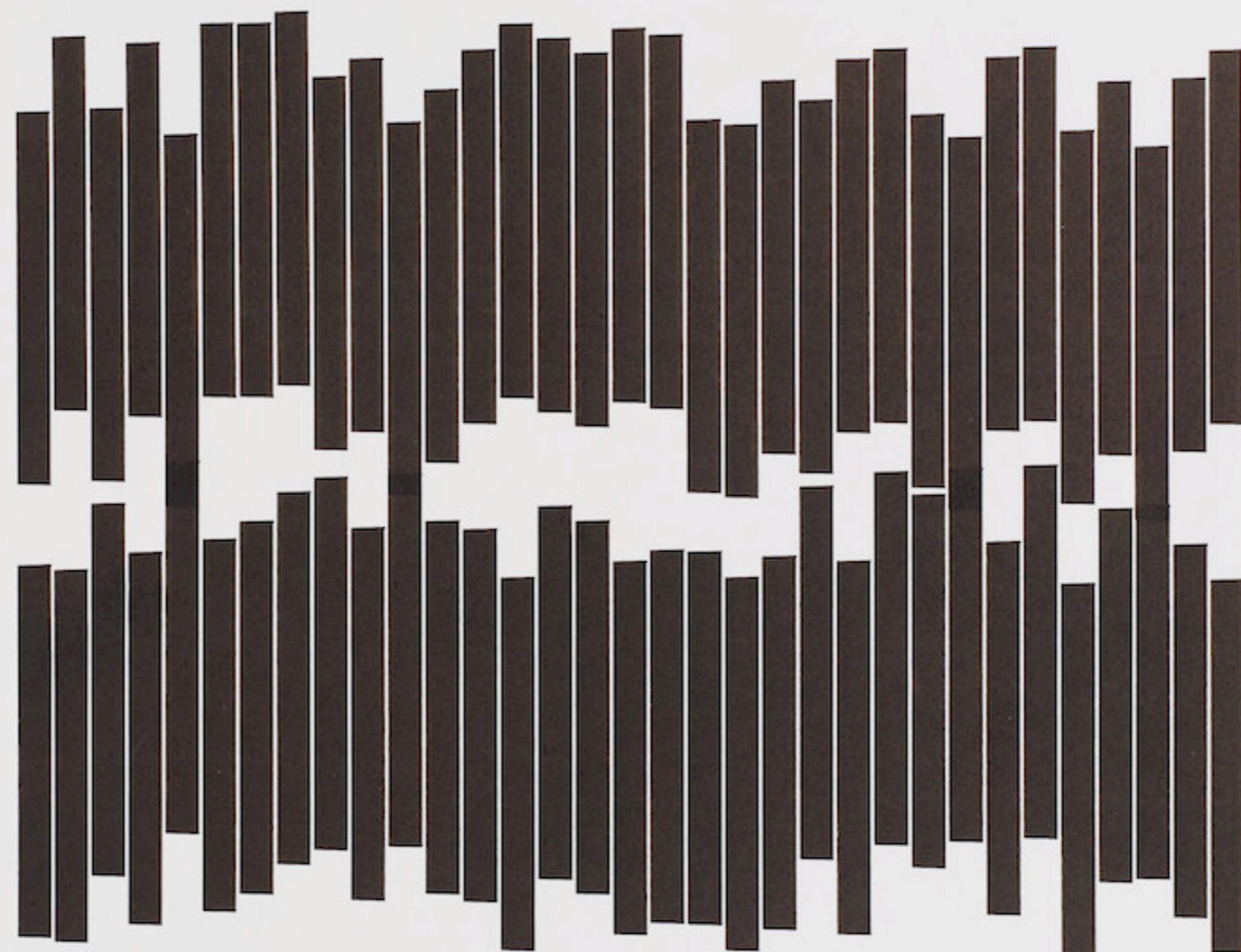
---

HISTORICAL EXAMPLES

# VERA MOLNAR

Vera Molnár (born 1924) is a French media artist of Hungarian origin. She is a pioneer of computer and generative art, active for over 75 years...





14/85

VERA MOLNAR

# Marcel Duchamp

## 3 Standard Stoppages

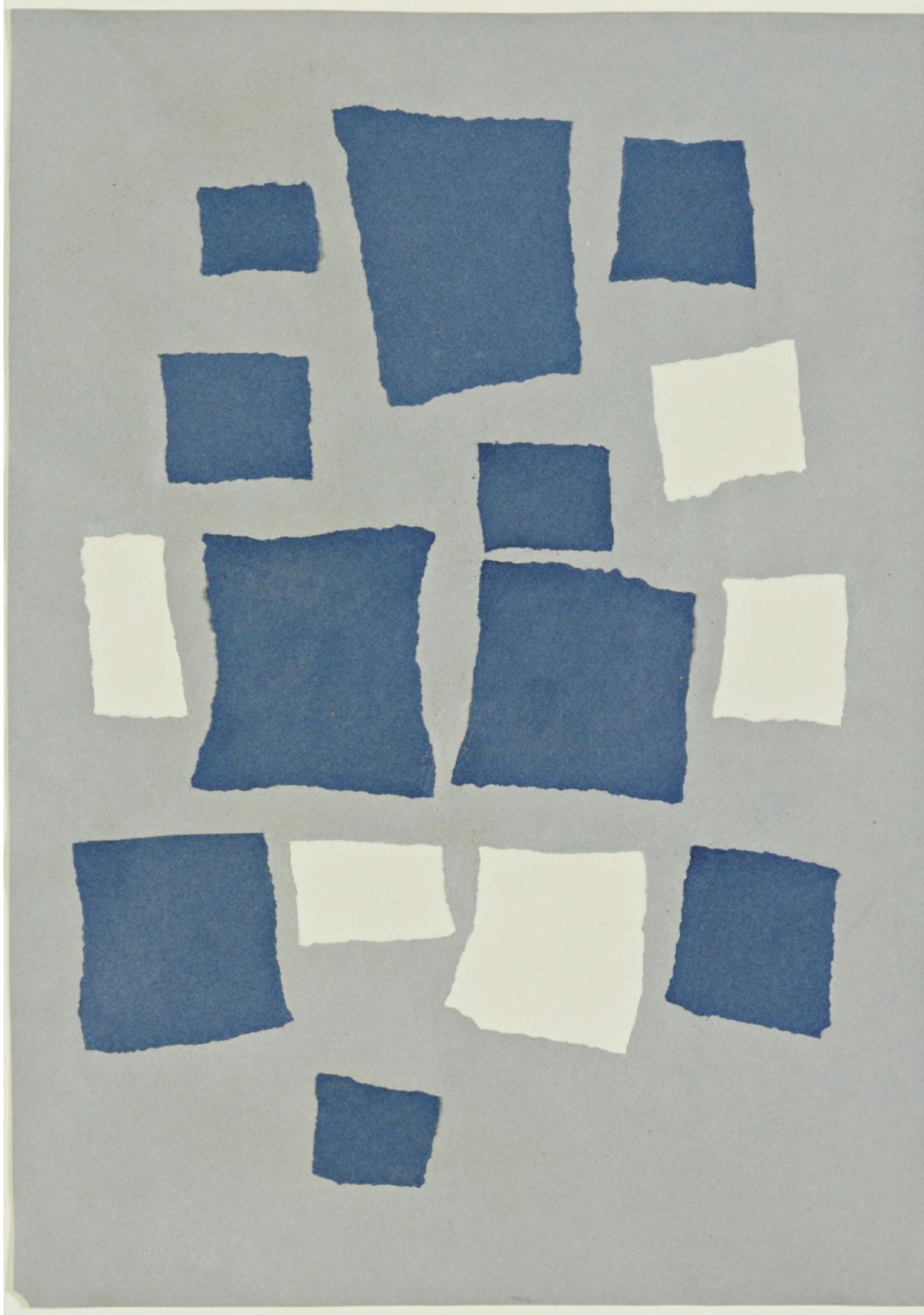
Paris 1913-14

In *3 Standard Stoppages* (3 *stoppages étalon*), Duchamp dropped 1-meter lengths of thread onto prepared canvases, one at a time, from a height of 1 meter. He then varnished them into place on the blue-black canvas strips and attached them to glass. He then cut three wood slats into the shapes of the curved strings, and put all the pieces into a croquet box. Three small leather signs with the title printed in gold were glued to the "stoppage" backgrounds.



# DADA

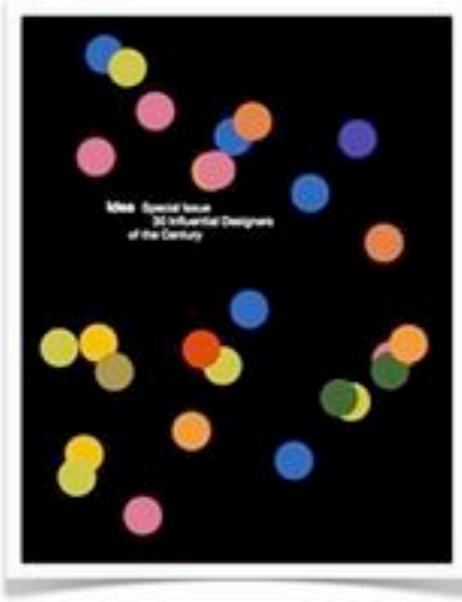
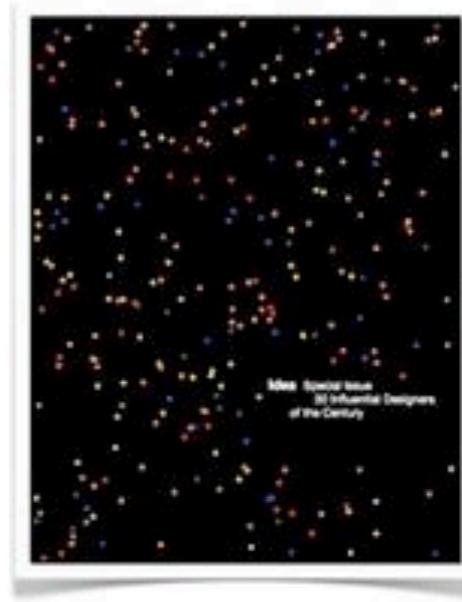
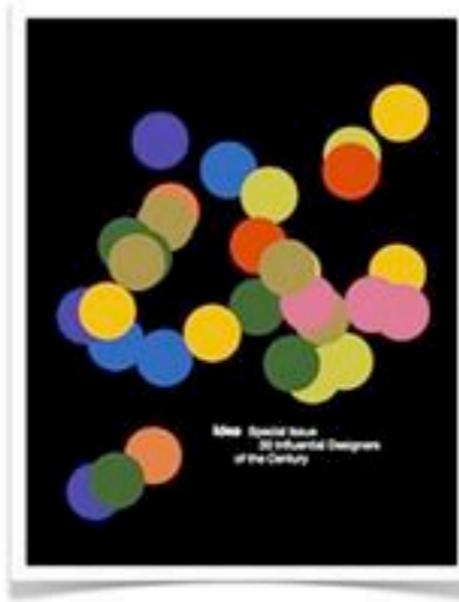
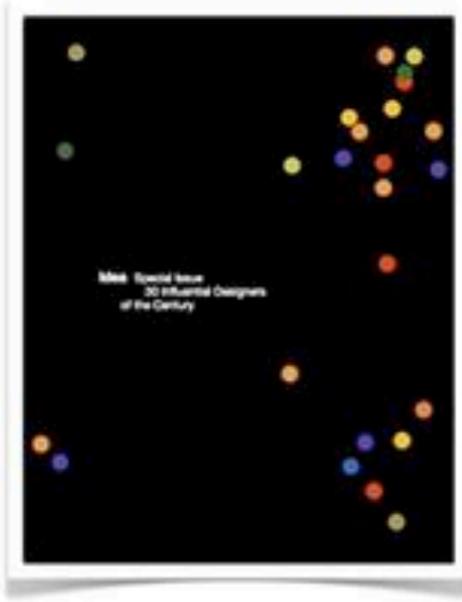
In this and similar works, Dadaist Jean Arp played with random composition by dropping painted pieces of paper onto a surface, then gluing them into place...



**Jean (Hans) Arp**

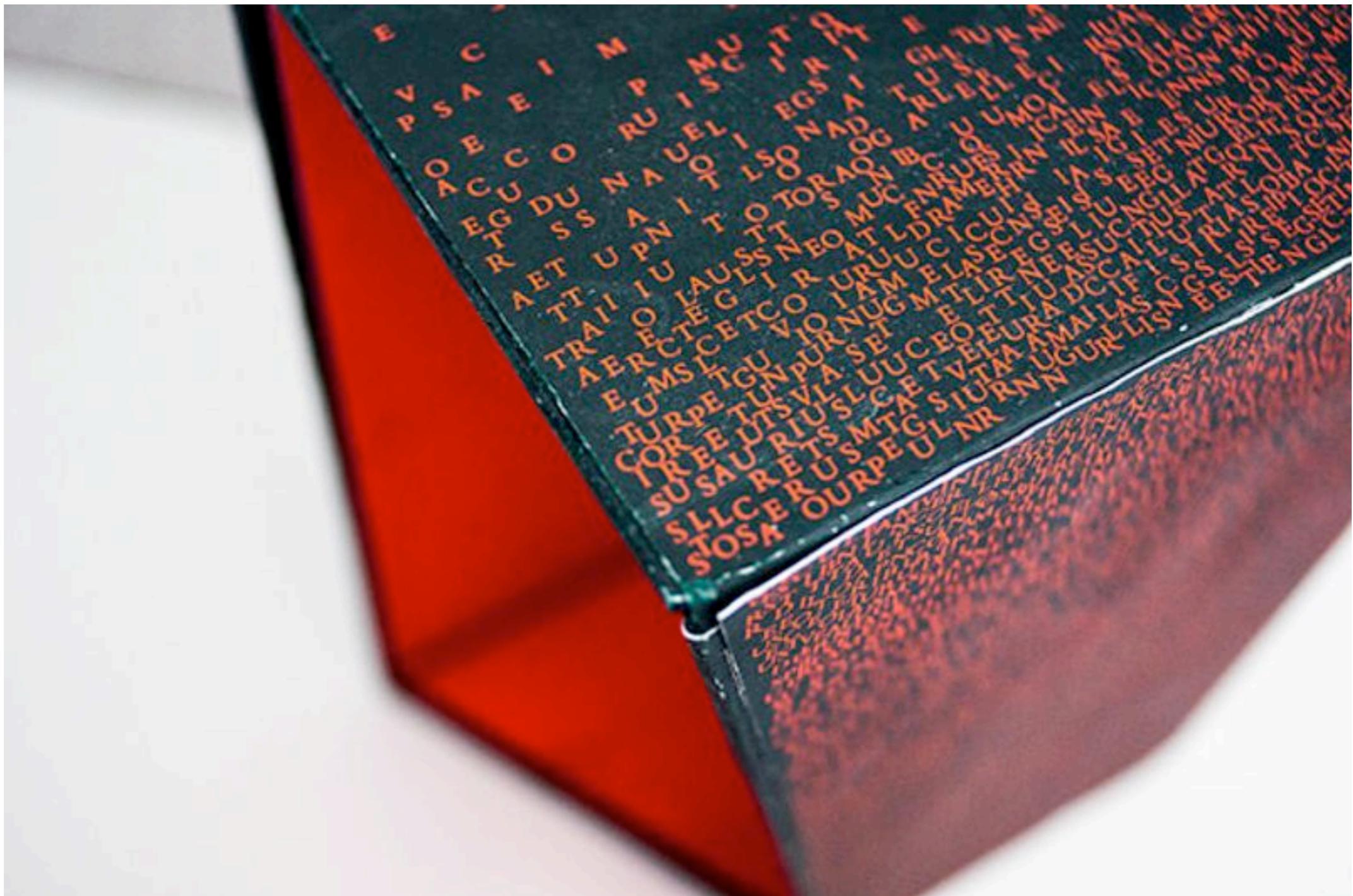
Untitled (Collage with Squares Arranged according to the Law of Chance)  
1916–17

# RANDOMNESS IN DESIGN



Paul Rand

# RANDOMNESS IN DESIGN



Design by Zhusi Xie, for a John Cage novel

# MUSIC OF CHANGES

## BY JOHN CAGE, 1951

### ARGUABLY THE FIRST SOUND COMPOSITION TO BE LARGELY DETERMINED BY RANDOM PROCEDURES



*Music of Changes* for solo piano, composed by John Cage in 1951 for pianist and friend David Tudor. The composition process involved applying decisions made using the *I Ching*, a classical Chinese text commonly used as a divination system. The *I Ching* was applied to sounds, durations, dynamics, tempo and densities.

# EMILY MASON

Abstract painter Emily Mason (1932–2019) used the laws of chance and gravity to create vital works with poured paint. She believed in chance operation, particularly as experienced in New York, where, she said, “you could be inspired by a tropical fruit in Chinatown and an exhibition on Byzantine art uptown...”



Emily Mason, *The Bullock Farm*, 1987, Oil on canvas, 52 x 42 inches.

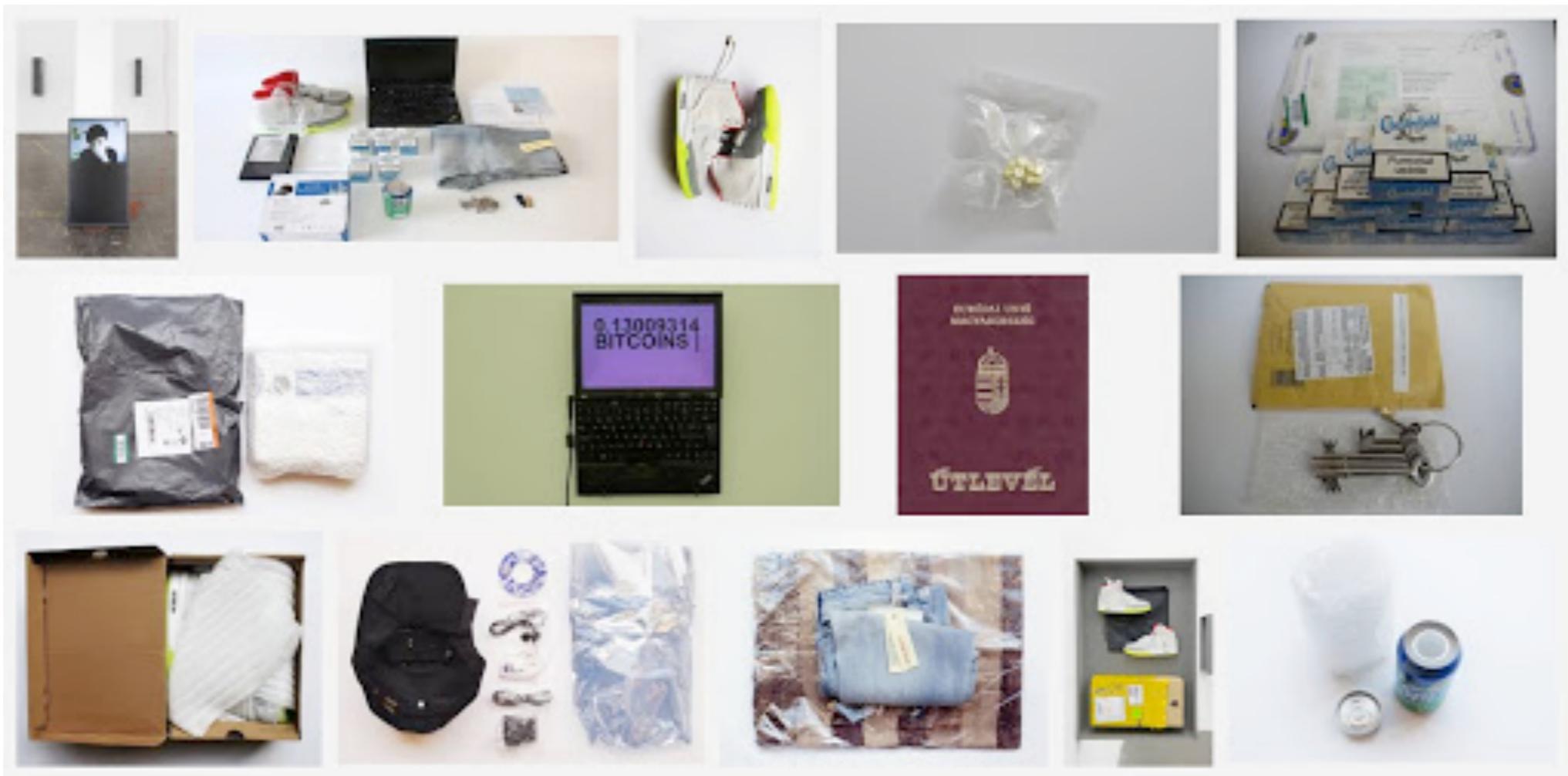
# CAROLEE SCHNEEMANN

Schneemann's video *American I-Ching Apple Pie* (1977) incorporated the notion of chance as a ruling cosmological principle, associated with the *I-Ching*, to free women from the confines of both the kitchen and the rational recipe.

Schneemann chose her cooking tools randomly, including colanders, strainers, nails, hammers, arrows, and ball bearings, using chance and Eastern philosophy to break down traditional gender boundaries.



# RANDOMNESS AS CRITICAL INTERVENTION



AN ALGORITHM WITH A WEEKLY BUDGET OF \$100 IN BITCOIN, RANDOMLY PURCHASES ITEMS FROM THE DARK WEB, INCLUDING ECSTASY, A HUNGARIAN PASSPORT, AND A BASEBALL CAP WITH A BUILT-IN CAMERA...

IN JANUARY 2015, THE SWISS POLICE CONFISCATED THE ROBOT AND ITS ILLEGAL PURCHASES. THEN THREE MONTHS LATER, RETURNED ALL (MINUS THE ECSTASY)

# CONCEPTUAL

---

WHY USE RANDOMNESS ?  
WHAT DOES IT MEAN ?

# WHY USE RANDOMNESS ?

- reduce authorial control
- avoid common habits of artist
- add variation to predictable outputs
- create surprise (for author or audience)
- explore a possibility space
- conceptual strategies

# **TECHNICAL**

---

# RANDOM()

## Description

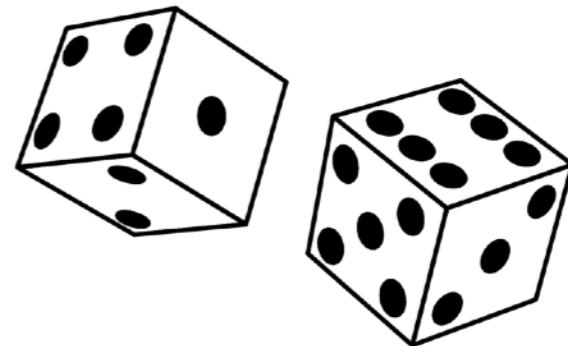
Return a random floating-point number.

Takes either 0, 1 or 2 arguments.

If no argument is given, returns a random number from 0 up to (but not including) 1.

If one argument is given and it is a number, returns a random number from 0 up to (but not including) the number.

If two arguments are given, returns a random number from the first argument up to (but not including) the second argument.



# RANDOM WALK

> sketch.js\*

Preview

```
1 let x = 200;
2 let y = 200;
3
4 function setup() {
5   createCanvas(400, 400);
6 }
7
8 function draw() {
9
10   x += random(-3, 3);
11   y += random(-3, 3);
12
13   background(255, 4);
14   noStroke();
15   fill(50);
16   circle(x, y, 10);
17 }
18
```

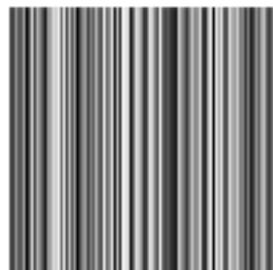


One of the most common techniques using randomness is the *random walk*, which shows up in a range of contexts, from the movement of financial asset prices, to the paths of particles in quantum physics...

# RANDOMSEED()

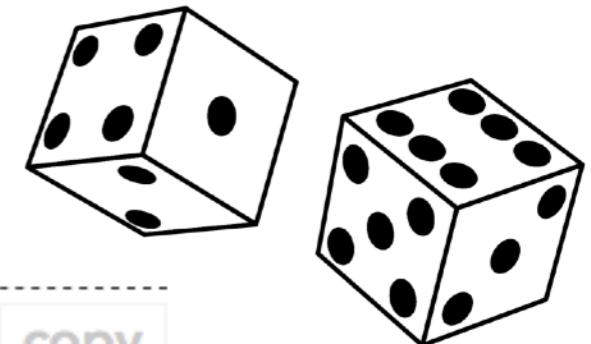
randomSeed()

## Example



```
randomSeed(99);
for (let i = 0; i < 100; i++) {
  let r = random(0, 255);
  stroke(r);
  line(i, 0, i, 100);
}
```

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



## Description

Sets the seed value for `random()`.

By default, `random()` produces different results each time the program is run. Set the seed parameter to a constant to return the same pseudo-random numbers each time the software is run.

**END**

DANIEL C. HOWE  
email: daniel@rednoise.org  
web: <https://rednoise.org/daniel>  
twitter/mastodon: @danielchowe