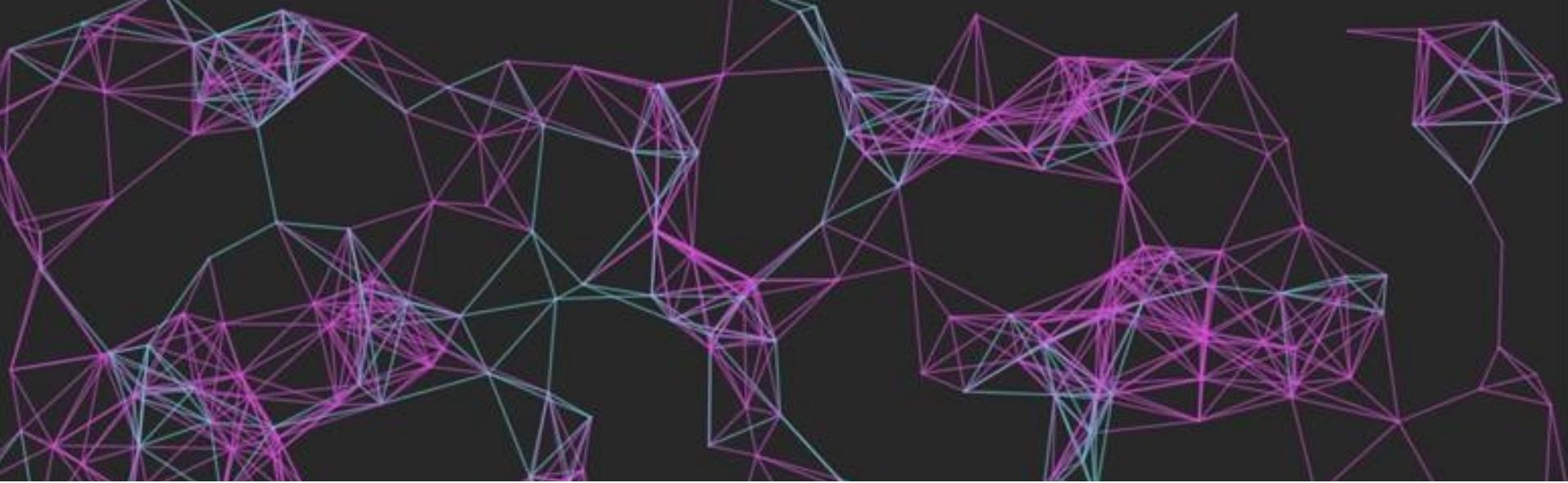


# **Creative Programming**

Friday 09:00am - 12:50am

Office Hours: Thursday 1pm - 2pm (online or via email)



Instructor: Tuba Ozkan



# Meeting

Who am I?

[www.tubaozkan.com](http://www.tubaozkan.com)

# Meeting

Who are you?

# Syllabus

<https://tinyurl.com/y7rkns5t>

# Course Materials

<https://github.com/ozkantuba/CreativeProgrammingI>

# **Class Blog ?**

What do you think about having  
a tumblr or wordpress blog?

# Github

-create an account now-

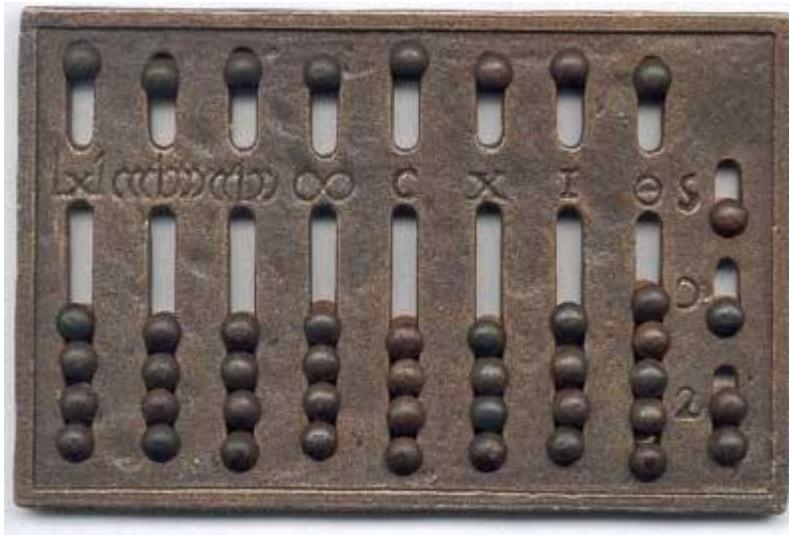
# **Creative Coding**

What is it?

**Creative Coding** is a branch of computer programming in which the goal is to **create something artistic + creative + playful** etc. instead of something that has a practical function.

# **Computing**

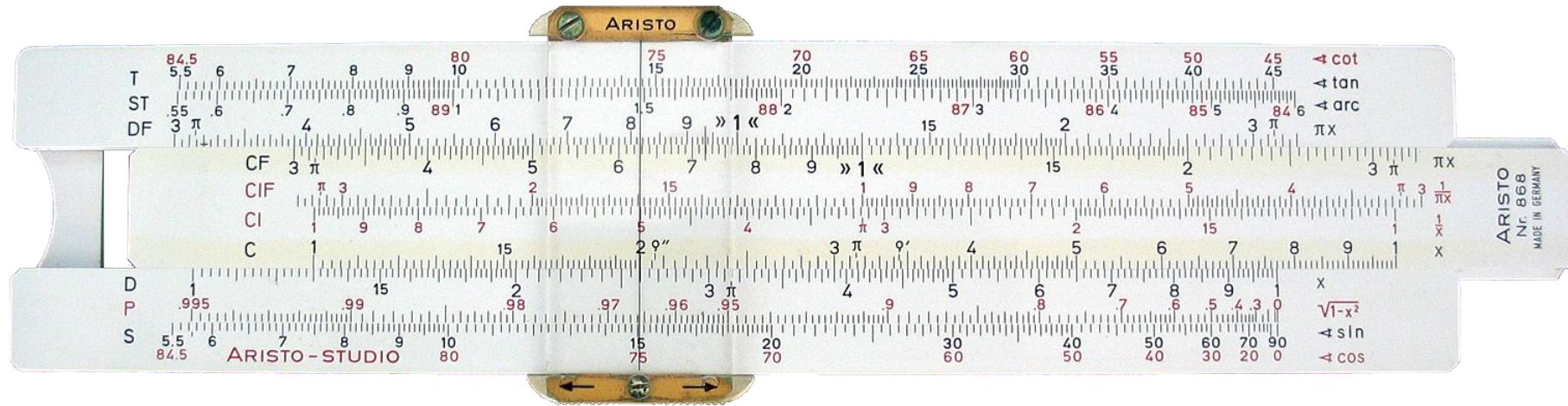
has been with us since ancient times



A modern replica of a Roman hand abacus from 1st century CE



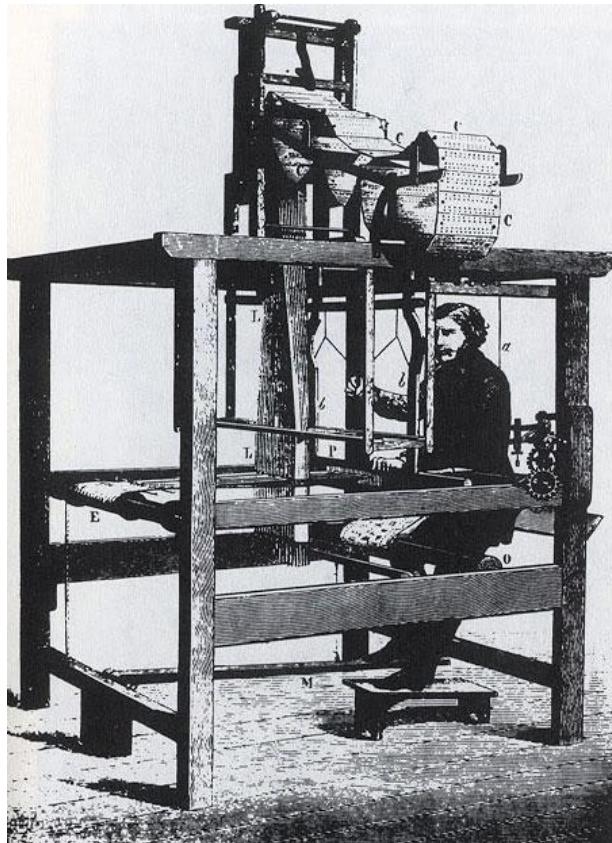
Antecedents to analog  
computers,  
the Antikythera mechanism



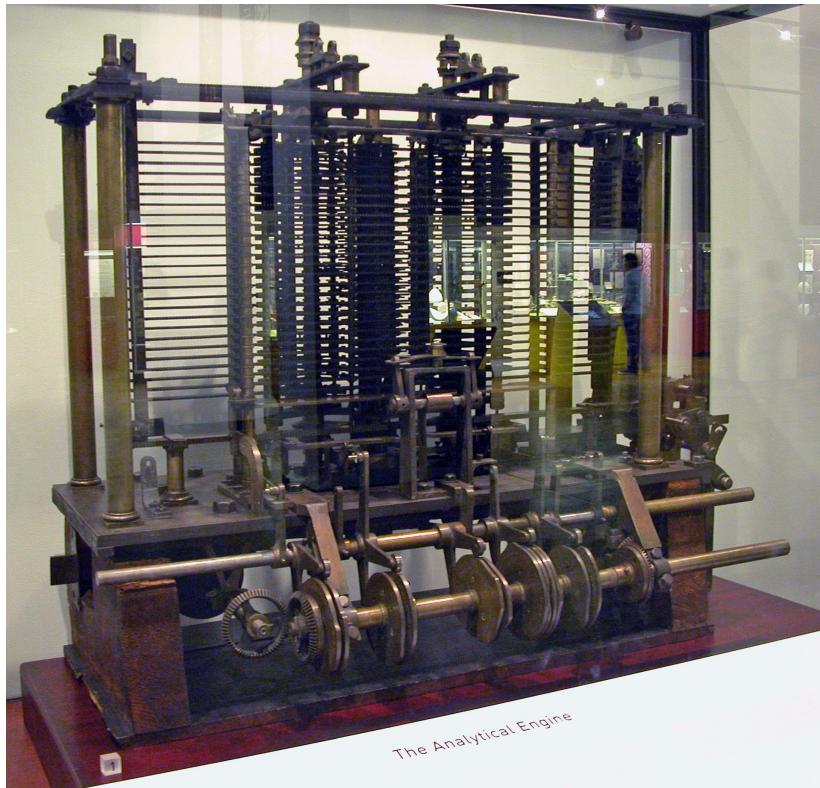
Early calculators, the slide rule in the 1600s



Pascal's mechanical calculator also from the 1600s



The Jacquard Loom, invented by Joseph Marie Jacquard in 1801, used instruction sets on punch cards to operate. (1800's)



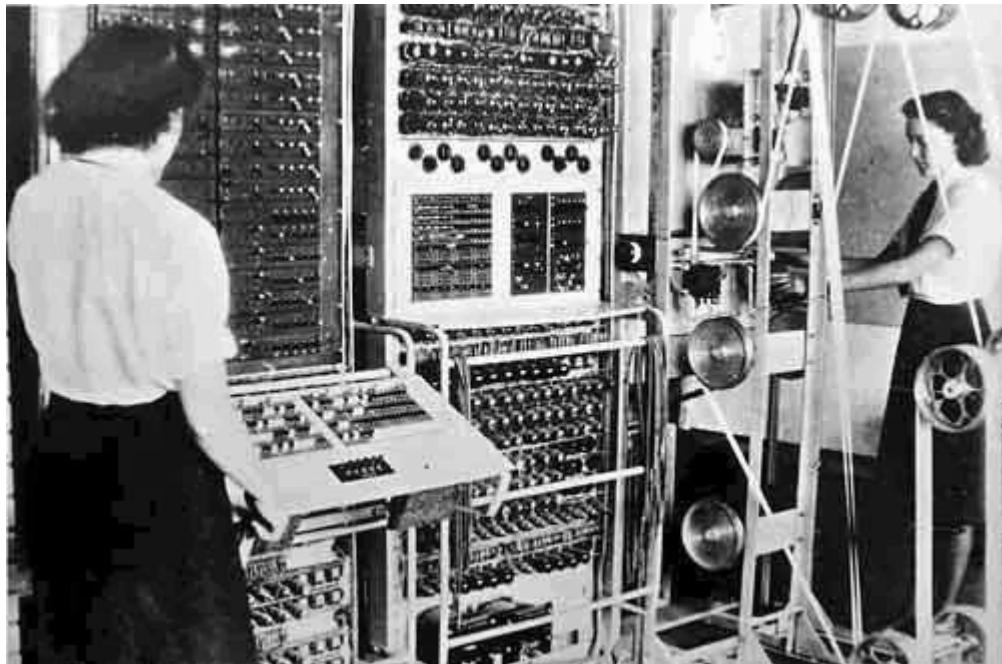
Charles Babbage's  
Analytical Engine, the  
first mechanical general  
purpose computer, 1833



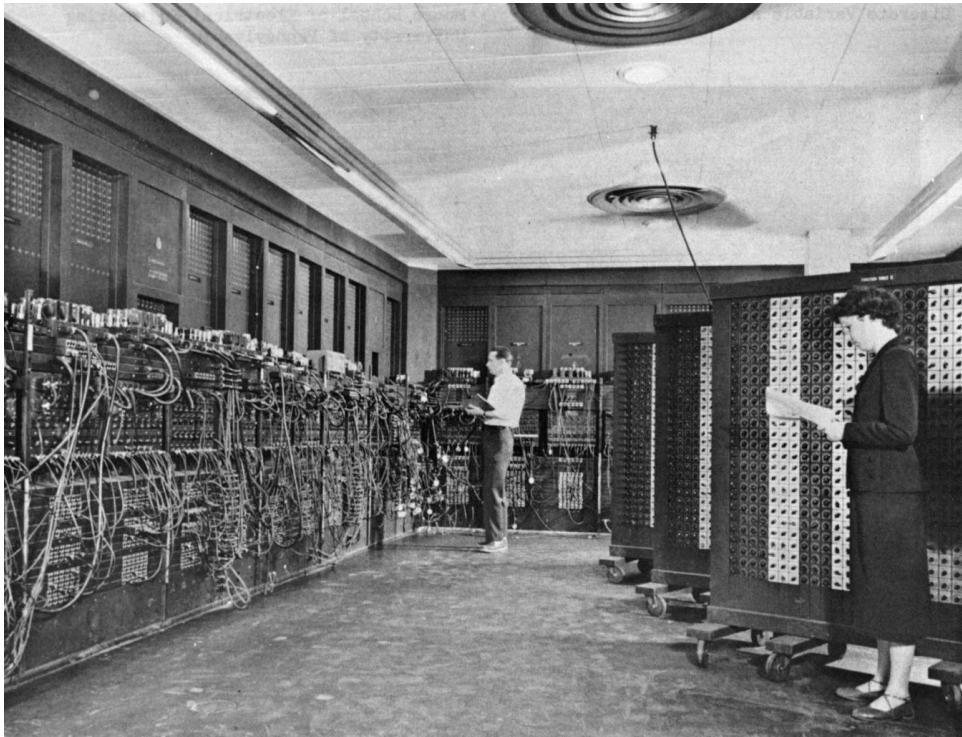
Astronomer Edward  
Charles Pickering's  
Harvard Computers,  
1925



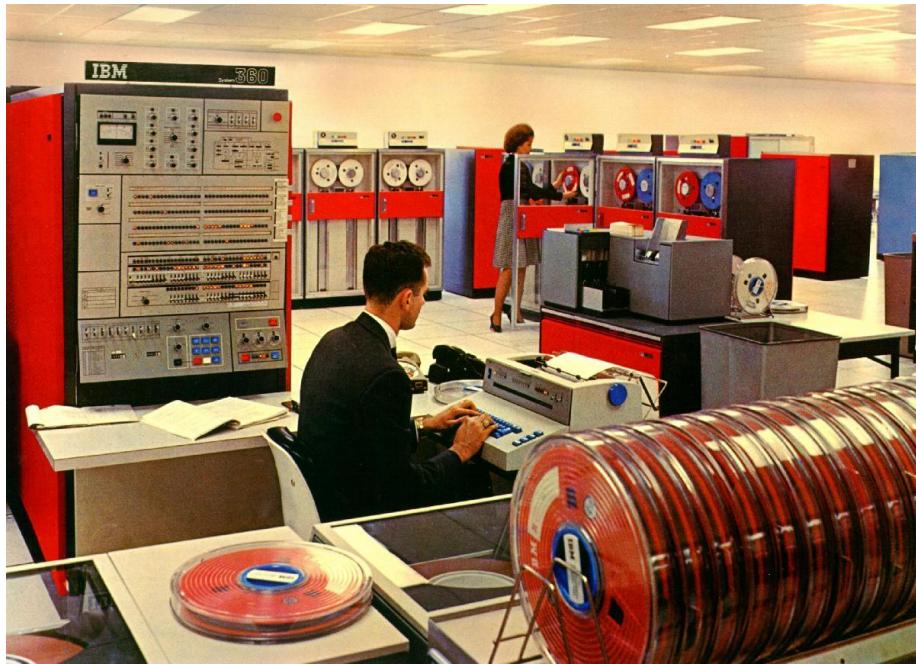
Differential analyser:  
A mechanical analogue  
computer designed to  
solve differential  
equations, 1942-1945)



Colossus: the first electronic digital programmable computing device. Used to break German ciphers during World War II, 1943.



ENIAC: the first Turing-complete electronic device, and performed ballistics trajectory calculations for the United States Army.



From mainframe computers  
-IBM 360 pictured-

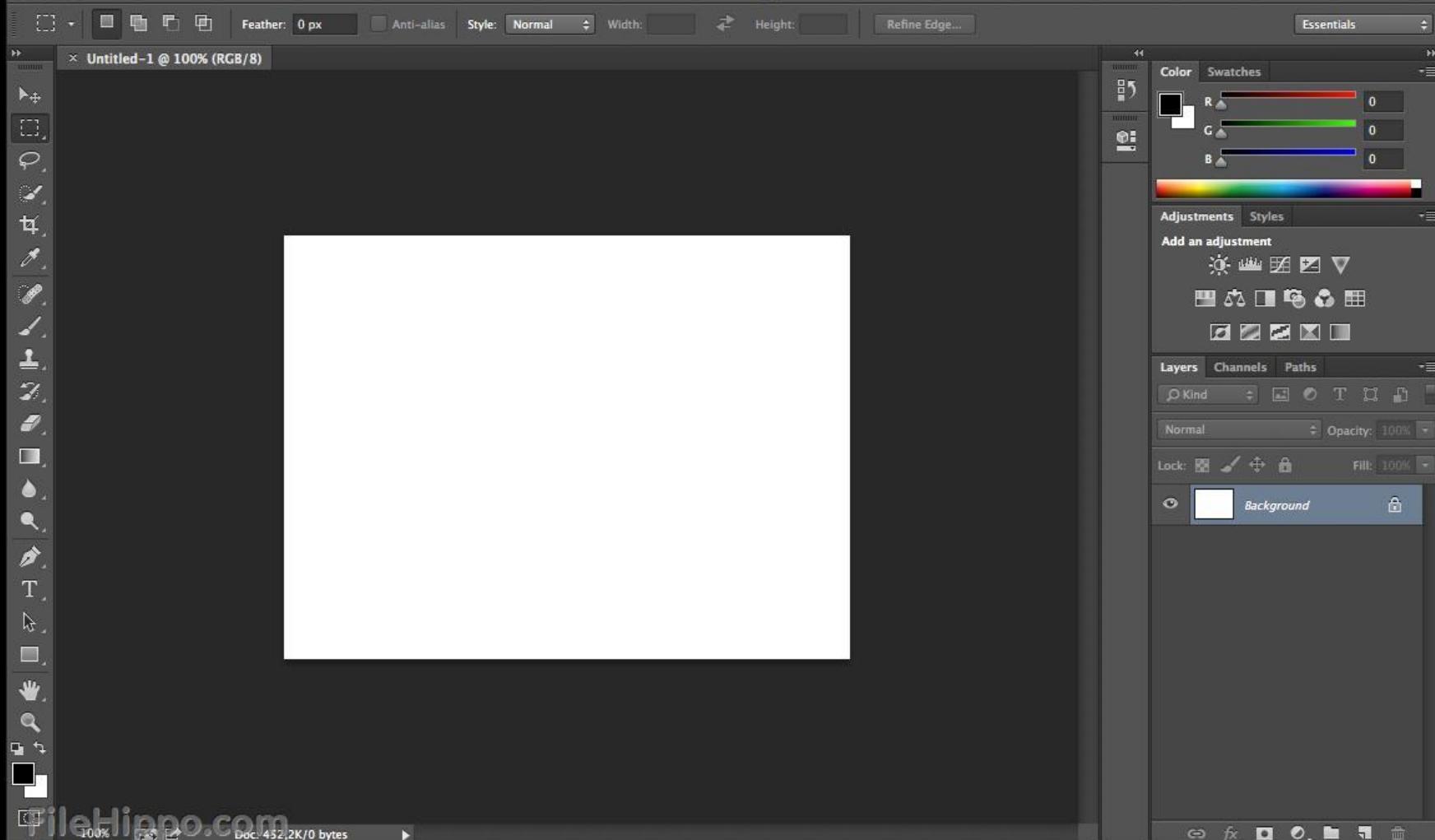


To personal computers...  
-Released 1981, MS-DOS-

**Why do we code?**



# Adobe Photoshop



# **What is code?**

a set of instructions

# **Translating Instructions for a Computer**

# What computers do:

**Store information** as 1s and 0s and  
**Perform math and logic operations** on it

Intermediate Languages and Libraries  
Expand the Computer's Vocabulary



Facebook ! 🔍

https://www.facebook.com

Search Facebook

Jonathan Home

Aggie Winsom shared フレッテ YOLO's photo.  
1 hr ·

when u smoke  


Like Comment Share

5 Write a comment...

Mohini Freya Dutta shared Kavita Krishnan's post.  
17 hrs ·

Omg dying

Amit Shah    
@AmitShah

Julia West likes Pete Bogdis's post in Northsiders SFCFC.

R Hunter Gough likes Joy Swiontek's post.

Alexandra Mathews was tagged in Plath/Hughes's photo.

Fernando Ausin-Gómez likes Ryan Luckey's post.

Mercedes Cossich likes Faithfull the

1 event invite

Martin Hunt and 1 other

TRENDING

- Reddit: Republican Presidential Nominee Donald Trump Participates in Website's 'Ask Me Anything'
- #TrumpSacrifices: Hashtag Emerges After Donald Trump Asserts He Has Made Sacrifices for America
- Ann Coulter: Commentator Calls Father of Late Soldier 'an Angry Muslim With a Thick Accent'
- Mitt Romney: Former Presidential Nominee Says He Thinks Donald Trump Could Win Presidency
- Hillary Clinton: Computer Systems Used by Presidential Candidate's Campaign Reportedly Hacked
- Stephen Hawking: Physicist Says How Wealth Is Understood Played a Crucial Role in Brexit
- North Korea: US 'Crossed the Red Line' by Putting Sanctions on Kim Jong Un, Country's Top Diplomat Says
- FBI: Agency Investigating Cyberattacks Against Democratic Party Groups and Clinton Campaign
- Roland S. Martin: Journalist Criticizes Rapper Bow Wow's Statement on Civil Rights Movement
- KRLD: Meteor  Chat (53) 

Squirtle / CP 278



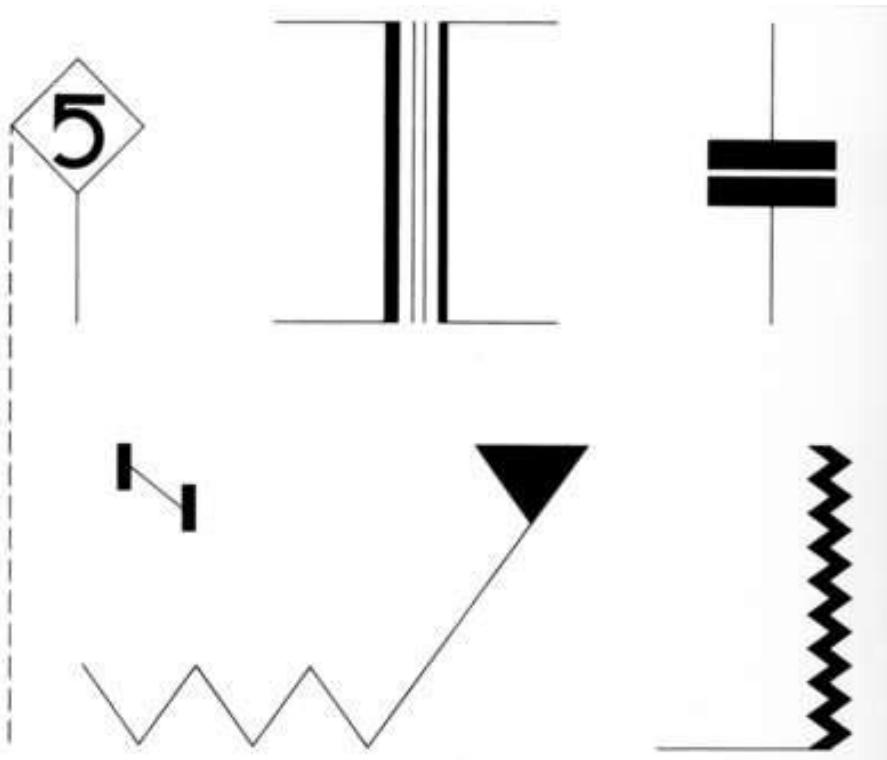
Why do we code?  
**to create cool stuff...**



Some pioneers in the field of  
generative / algorithmic art



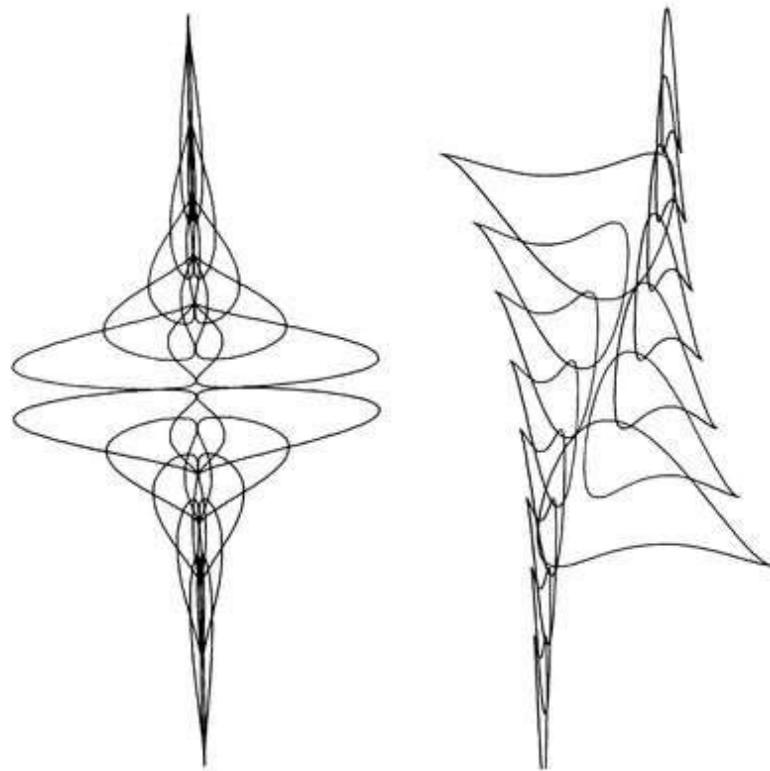
**Manfred Mohr** At His Plotter



**Manfred Mohr**

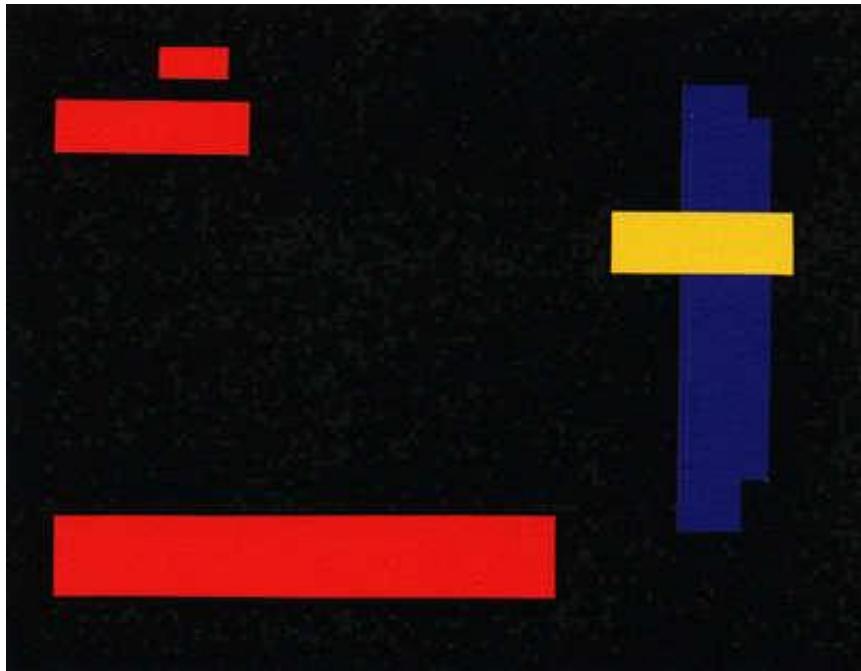
(non-algorithmic)

Bild 16/469 ohne Titel  
1969



**Herbert W. Franke / Peter Henne**  
Algebraische Kurven  
1969

<https://bitforms.art/artists/mohr>



**Jan Baptist Bedaux / Jeroen  
Clausman / Arthur Veen / Compos 68**

Compos 68 Series II  
1969

other, c., computer-generated  
Ink, felt-tip pen, paper  
3.88 x 28 cm

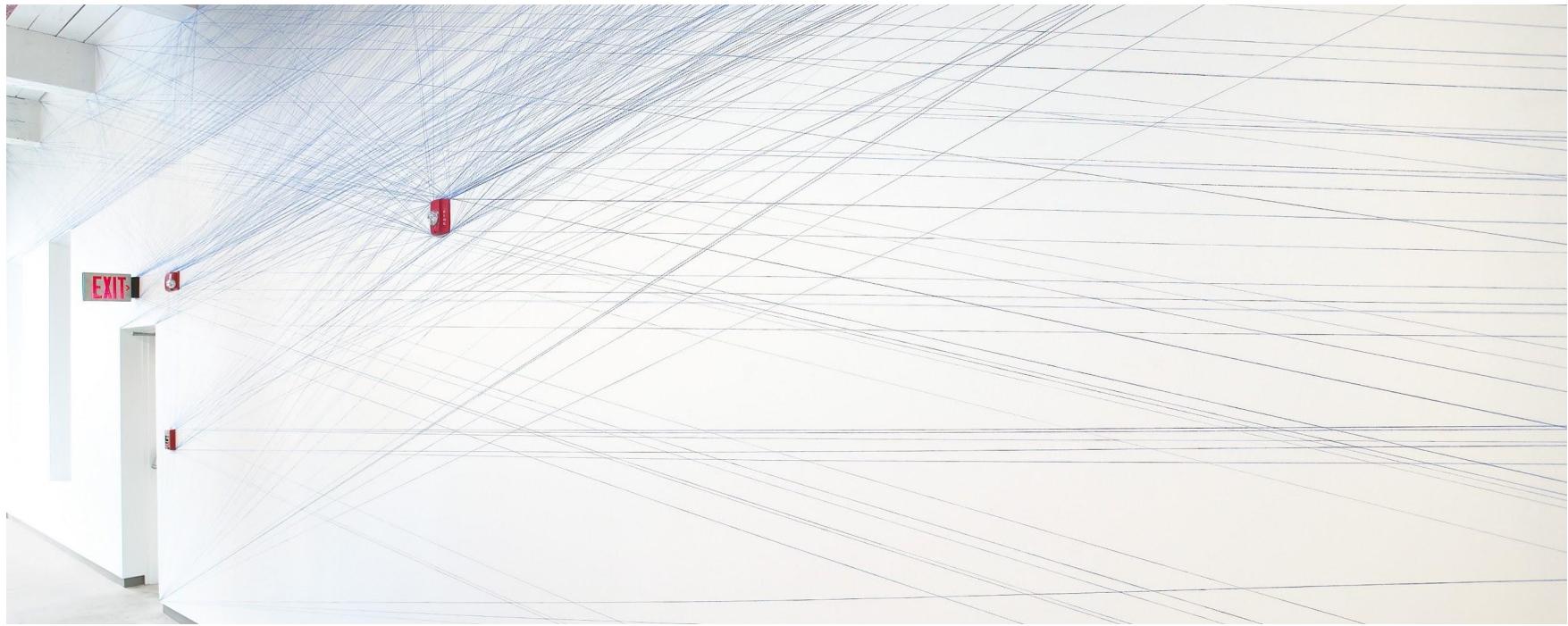


**Sol Lewitt**

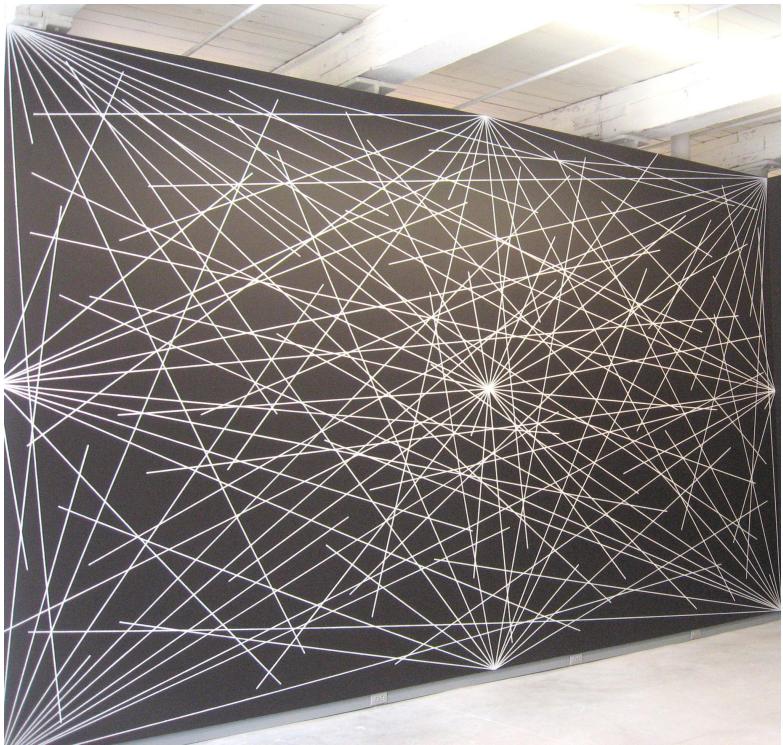
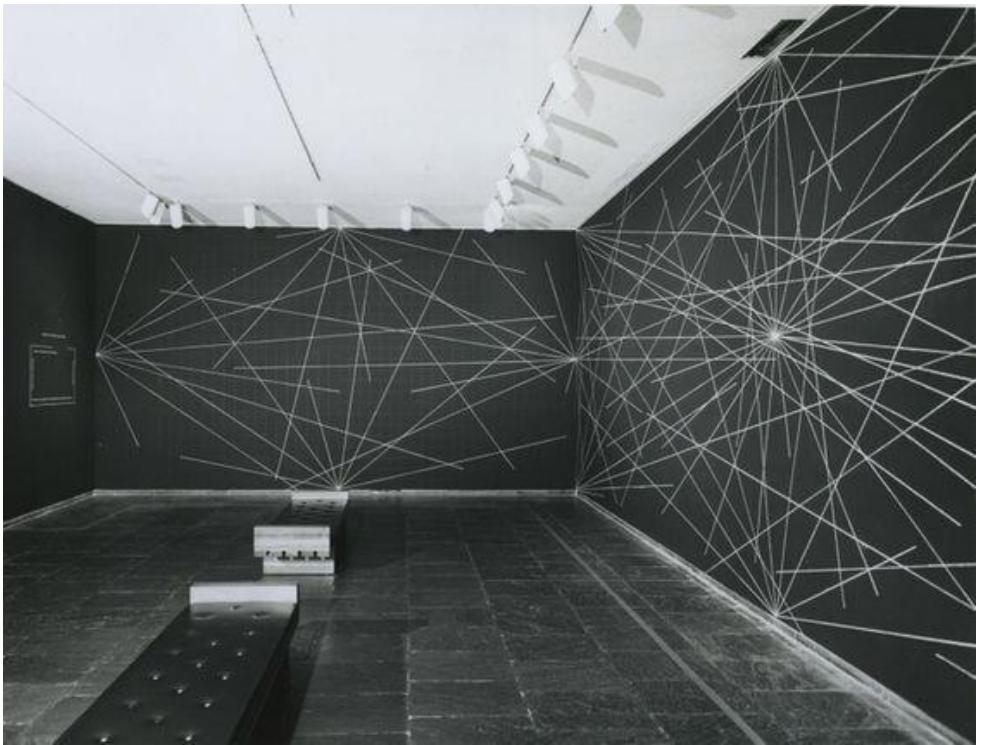
Six Geometric Figures Plus Two  
1980-81



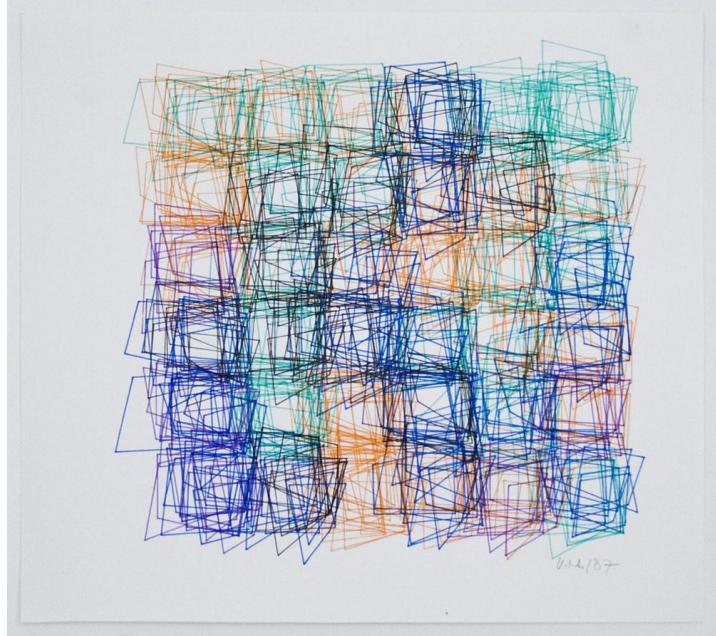
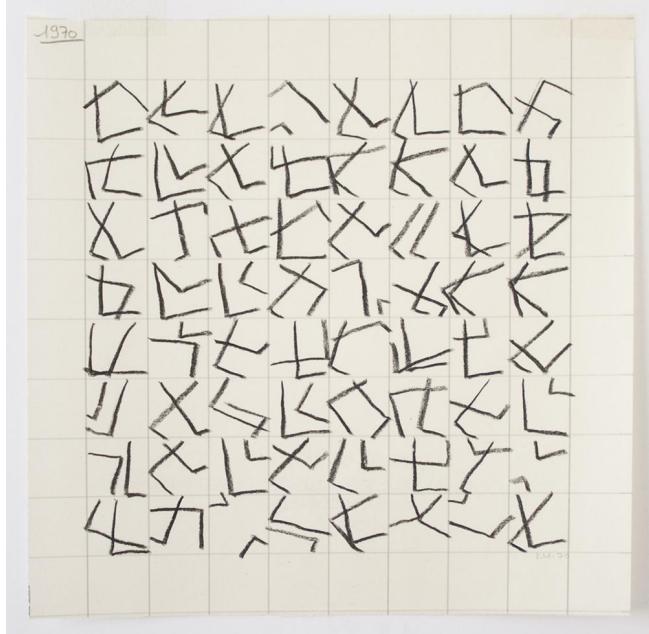
**Sol Lewitt**



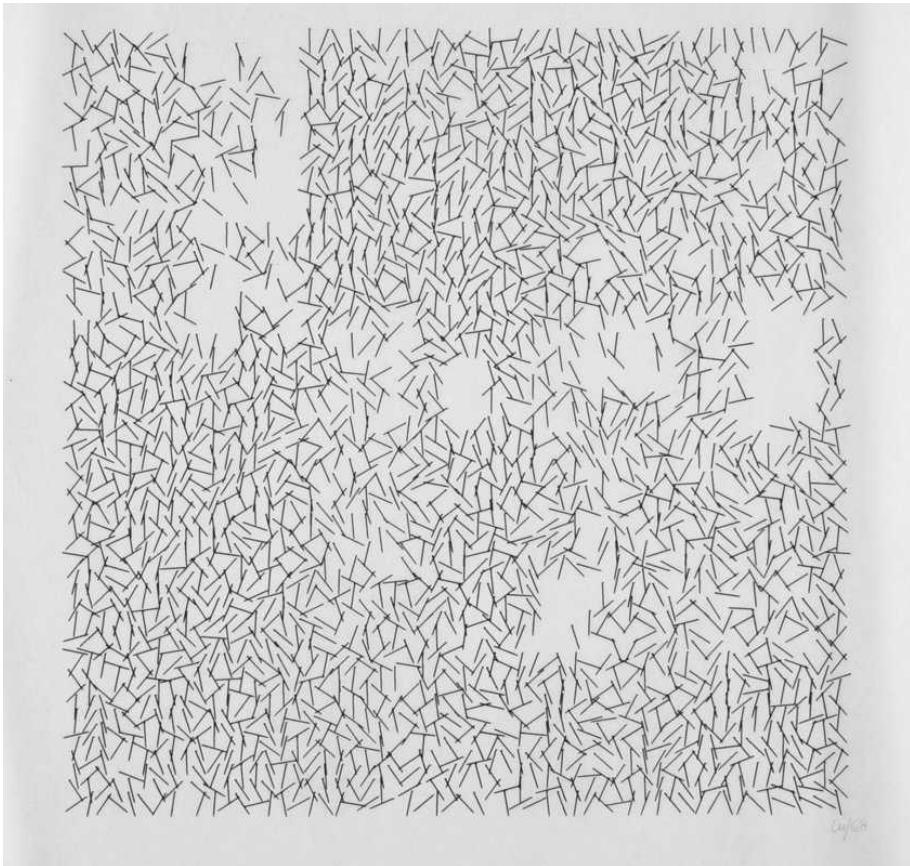
**Sol Lewitt**



**Sol Lewitt**



**Vera Molnar**  
Square  
Structures  
1987



**Vera Molnar**  
Interruptions  
1968-69

**Today?**

**Break Time**

Processing  
openFrameworks  
TouchDesigner  
Cinder

Vvvv  
SuperCollider  
Pure Data  
Max MSP  
GNU Octave...



# S W E L L

---

Luke Twyman



# CRUSH

---

Luke Twyman



Luke Twyman



Luke Twyman

# W A V E

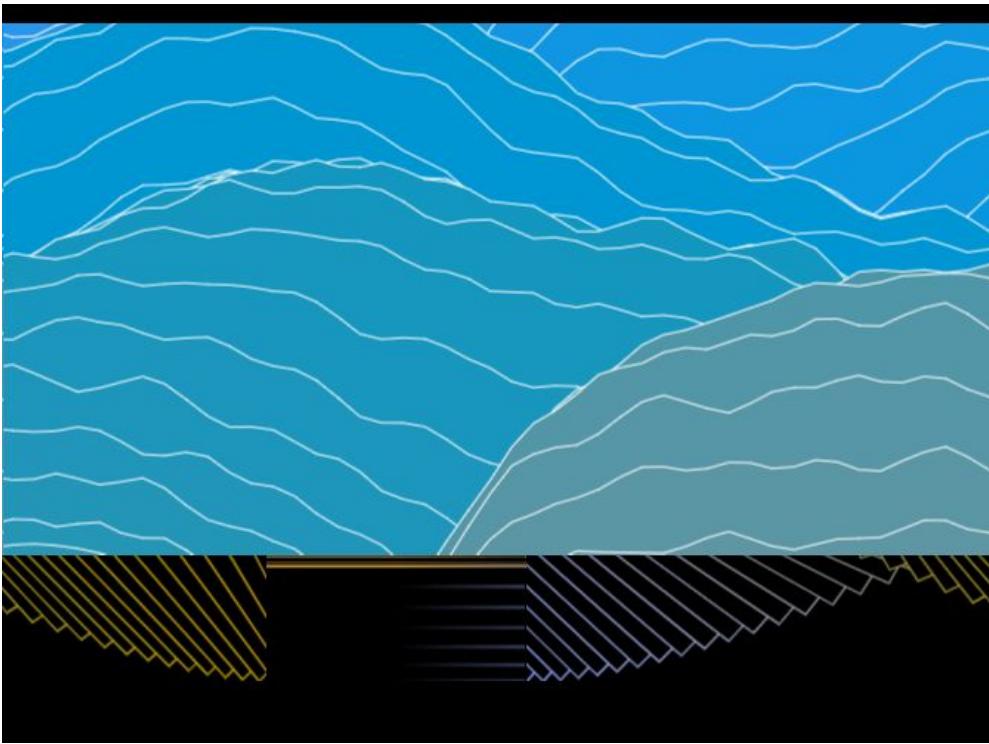
---

Luke Twyman

# VOICELESS UNIT

T X T

Luke Twyman



Centiscript: A tool for visual  
thinking by Satoshi HORII

Click link:

<http://ex.rzm.co.jp/centiscript/ss.html>

Toxiclibs 2010 Showreel [Processing]

<https://vimeo.com/15379147>

V...Mir Live you Play 2017

<https://vimeo.com/217668746>

Mill Touch 'Behind the Scenes'

<https://vimeo.com/27718212>

Urban Futures

<https://vimeo.com/35478478>

Cell Cycle

<https://vimeo.com/14835105>

Kuflex exhibition (OF)

<https://vimeo.com/224376237>



# Processing

What is it?

Processing is an **open-source graphical library** and integrated development environment (IDE) / playground built for the **electronic arts, new media art, and visual design** communities with the purpose of **teaching non-programmers the fundamentals of computer programming** in a visual context.

Processing uses the **Java language**.

The project was initiated in 2001 by Ryan Hopkins and Casey Reas and Ben Fry. In 2012, they started the Processing Foundation along with **Daniel Shiffman**, who joined as a third project lead.

# Download Processing

<https://processing.org/>

How do we communicate  
with computers?

# Programming Concepts

# Language Analogy

Variables  
Control Structures  
Data Structures  
Syntax  
Tools

# Syntax

The **syntax** of a programming language is the **set of rules** that define the combinations of symbols that are considered to be correctly structured programs in that language.

+Think about the syntax of  
a regular language like English, Spanish etc.

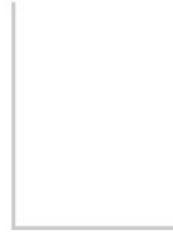
# Variables

```
int count;
```

Type



Name



# Type of Variables

## **int**

- stores an integer (eg. 1)

## **float**

- stores a number with a decimal point (eg. 9.31)

## **String**

- stores text (eg. "Bootcamp 2016")

## **boolean**

- true/false

# Control Structures

A **control structure** is a block of programming that **analyzes variables** and **chooses a direction in which to go based on given parameters**. The term *flow control* details the direction the program takes (which way program control “flows”). Hence it is the **basic decision-making process in computing**; flow control **determines how a computer will respond when given certain conditions and parameters**.

- +Function
- +If statements
- +For Loops

# **Functions**

A **function** is a type of procedure or routine

## For Loops

A **for-loop** (or simply **for loop**) is a control flow statement for specifying iteration, which allows code to be executed repeatedly

# Pseudocode

is an informal high-level description of the operating principle of a computer program or other algorithm.

## Pseudocode 1

```
// spread peanutButter on breadSlice1  
// spread jelly on breadSlice2  
// put bread slices together
```

```
var breadSlice1;
var breadSlice2;
var peanutButter;
var jelly;

function main() {
    spreadOnBread(peanutButter, breadSlice1)
    spreadOnBread(jelly, breadSlice2)
    putBreadTogether(breadSlice1,breadSlice2)
}
function spreadOnBread(a,b) {
a goes on top of b
}
function putBreadTogether(x,y) {
x + y
}
```

# In-Class Activities

# Pseudocode: Your Turn!

pair up with a friend and think of some simple everyday activity  
then identify the variables and functions you would break it down into.

# Start Processing

## PseudoCode 3

```
//create a variable and assign a word which is a name  
//create a variable and assign a word which is a verb  
//create a variable and assign a word which is an another  
name  
  
//create a variable and assign a number  
//create a variable and assign a float number  
  
//print some sentences by using the words and numbers stored  
in your variables
```

# **Drawing Exercises**

Team-up as 3-4

1. Each member of your group should draw a dot on the page, one at a time, in a point that is as far from other dots as possible. Repeat many times and as quickly as possible.

2. Draw one straight line on the page of any length and angle. Each person should draw another straight line off from the end of an existing line, plus add a new straight line somewhere else on the page. No lines should touch any other. Repeat.

3. Draw a blob somewhere on the page. Each person should draw a new blob that intersects with another blob and doesn't touch any others. Repeat.

4. Draw a curving line in from the edge of the paper. Each person should draw another curving line from the edge, not touching any other lines. Repeat.

5. From Sol LeWitt (Our twist: draw in one square, then pass to the person on your right.)

`WORK FROM INSTRUCTIONS (1971):`

`USING A BLACK, HARD CRAYON DRAW A TWENTY INCH SQUARE. `

`DIVIDE THIS SQUARE INTO ONE INCH SQUARES. WITHIN EACH`

`ONE INCH SQUARE, DRAW NOTHING, OR DRAW A DIAGONAL`

`STRAIGHT LINE FROM CORNER TO CORNER OR TWO CROSSING`

`STRAIGHT LINES DIAGONALLY FROM CORNER TO CORNER. `

6. Draw a shape and draw a series of evenly-spaced points along the edge, about 20-30 of them. Pass the person on your right. On this new sheet, write a set of simple instructions about what to do from these points. Pass to the right again and carry out the instructions on this new sheet.

# Assignment

Read Manfred Mohr's *\*Computer Graphics\** essay

By using it and our in-class work as inspiration, create a more involved **drawing** (min. 8.5x11 inch paper) using instructions you create.