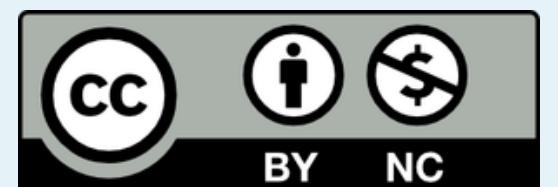


コードとデザイン

東京藝術大学 芸術情報センター開設科目 金曜4-5限 第1週

2024.04.12 松浦知也 (matsura.tomoya@noc.geidai.ac.jp teach@matsuuratomo.ya.com)



Google Classroom

rzggu2e



今日の予定

- オリエンテーション(20min.)
- Conditional Design ワークショップ (50min.)→ラボに移動します！
- 履修上の注意点など説明 (15min.)

自己紹介

- 神奈川県茅ヶ崎市出身(1994)
- 東京藝術大学音楽環境創造科卒(2017)
- 九州大学大学院芸術工学府 修士、博士後期課程(2017~2022)
- 東京藝術大学芸術情報センター 特任助教(2022~)
- YCAM(Intern,2015),teamlab(2015~2017),School for Poetic Computation(2018)



photo: Takehiro Goto

電子楽器を作って演奏したり、展示のための音プログラミングをしたり、音のためのプログラミング言語を作ったりしています。

<https://matsuuratomoaya.com>

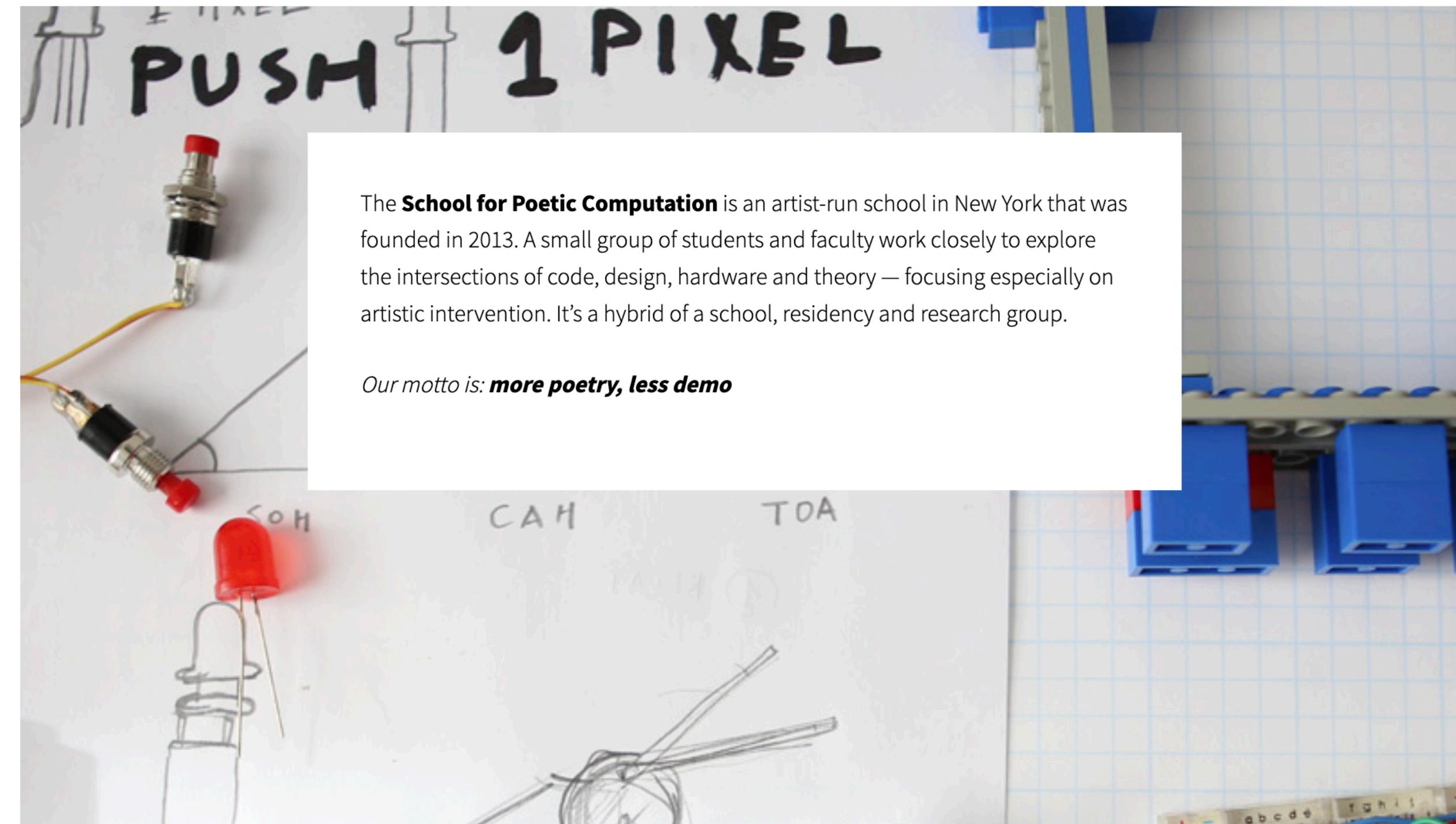
School for Poetic Computation

School for Poetic Computation

Participate:
Fall/Winter 2021-22
COCOON Summer
2021
Our Transition

Mission
People
Classes
Press
FAQ

Instagram 
Twitter 
Newsletter 
Blog
Finances



The **School for Poetic Computation** is an artist-run school in New York that was founded in 2013. A small group of students and faculty work closely to explore the intersections of code, design, hardware and theory — focusing especially on artistic intervention. It's a hybrid of a school, residency and research group.

*Our motto is: **more poetry, less demo***

<https://medium.com/sfpc/sfpc-fall-2018-1%E9%80%B1%E7%9B%AE-6fc844be3d83>



SFPC Summer 2019 in Yamaguchi <https://vimeo.com/363716118>
<https://potari.jp/2019/12/3673/>

Code and Design

Code and Design

Generative Design

Computational Design

Creative Coding

...

この授業で取り扱う内容

- ・コンピューターの基礎的な仕組みを実際に作りながら理解する
 - ・含：電子工作で簡単なコンピューターを作る
- ・コンピューターの多様なインターフェースのひとつとしてデジタルファブリケーション技術の基礎を身につける
- ・プログラミングとハードウェアを組み合わせた環境（Arduino）に慣れる
- ・芸術あるいは社会の中でコンピューターを使うことに意識的になる

参考：後期の「メディアアート・プログラミングII」ではもっとソフトウェア寄りの内容を扱います！

この授業で（直接は）取り扱わない内容

- VR
- AI
- NFT

Code and Design

規範 と デザイン

コードが法になる



“Code version 2.0” ローレンス・レッシグ著、山形浩生訳、翔泳社（2007）



今すぐ始める おすすめ リソース ヘルプ

ログイン

アクセス

再生数を正当に獲得したにもかかわらず、不正操作を理由にコンテンツが削除されたとお考えの場合は、ディストリビューターまたはレーベルに連絡し、当該コンテンツを正当にプロモーションするために採用された方法について情報を確認してください。ディストリビューターまたはレーベルとSpotifyのチームは連携し、迅速に問題を解決できるよう努めます。

ストリーミングの人為的操縦へのSpotifyの対策

人為的に操作されたストリーミングとは、ユーザーの真の再生インテントが反映されていないストリーミングのことです。これには、自動化されたプロセス(ボットやスクリプト)を使ってSpotifyを操作しようとする行為がすべて含まれます。

Spotifyは、エンジニアのリソースを投入し、Spotifyにおけるストリーミングの人為的操縦を検出、抑制、排除するための研究を行っています。これは、アーティストたちに芸術で生計を立てる機会を提供するというSpotifyのミッションを円滑に遂行し、権利者が可能な限り正当な利益を得られるようにするものです。不正なストリーミングがあれば、正直かつ懸命に活動しているアーティストが影響を受けるということであり、これはSpotifyにとって妥協できない非常に重大な問題です。

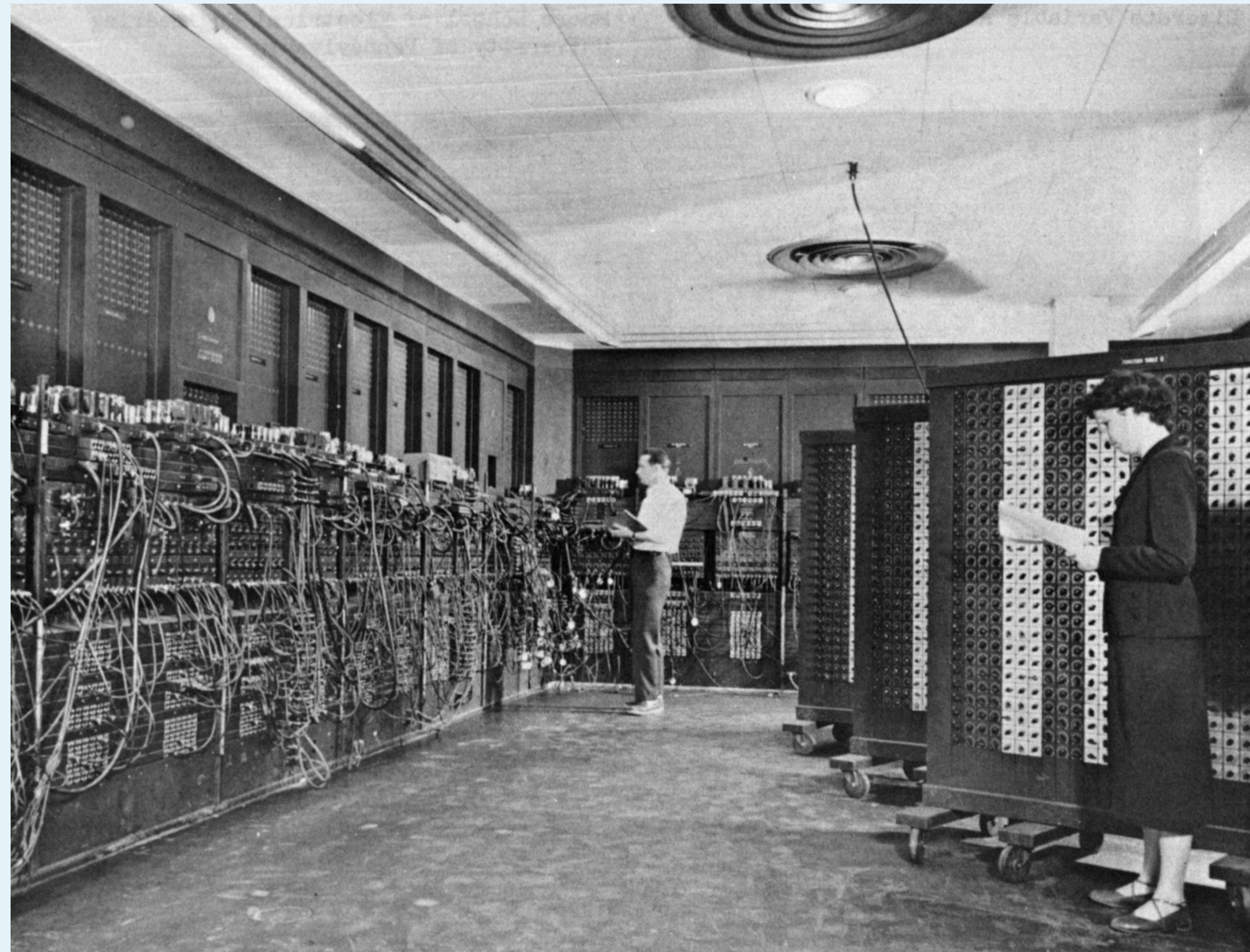
<https://support.spotify.com/jp/artists/article/third-party-services-that-guarantee-streams/>

あなたは制作にコンピューターを使っていますか？

あなたは制作にコンピューターを**主体的に**使っていますか？

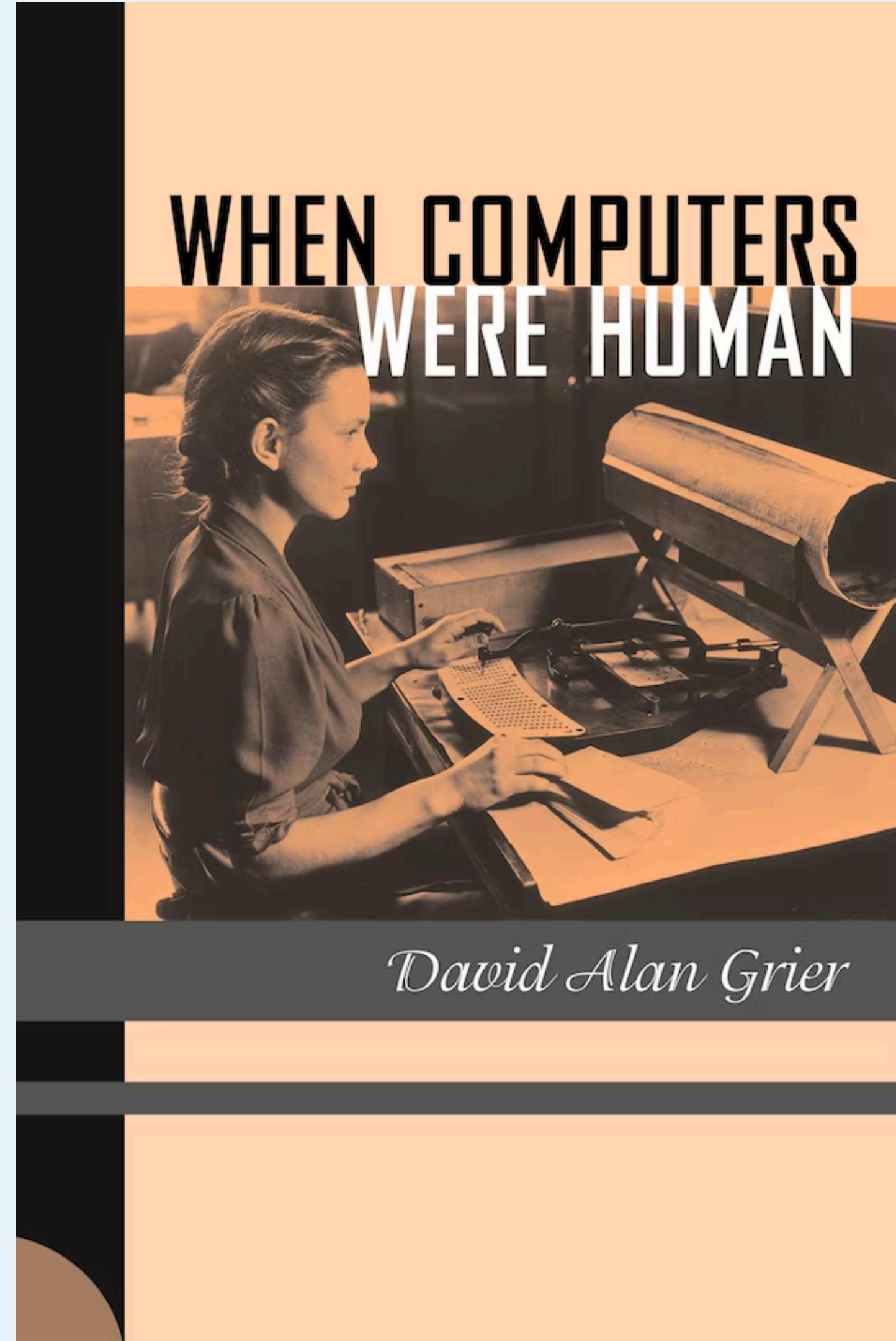
Q：コンピューターはいつ誕生しましたか？

1. 1940年代
2. 1950年代
3. 1960年代
4. 1970年代



ENIAC(1946)

Public Domain: <https://commons.wikimedia.org/wiki/File:Eniac.jpg>



“When Computers Were Human” by David Alan Grier (2005)
Princeton University Press, 9780691133829

キヤノンMJがお届けする安全なデジタル活用のためのセキュリティ情報
サイバーセキュリティ 情報局



トピック別 ▾

立場別 ▾

キーワード事典

ホワイトペーパー ▾

セキュリティ注意喚起 ▾

サイバーセキュリティラボ ▾

コンプライアンス、教育 企業ユーザー向け

2016.4.13

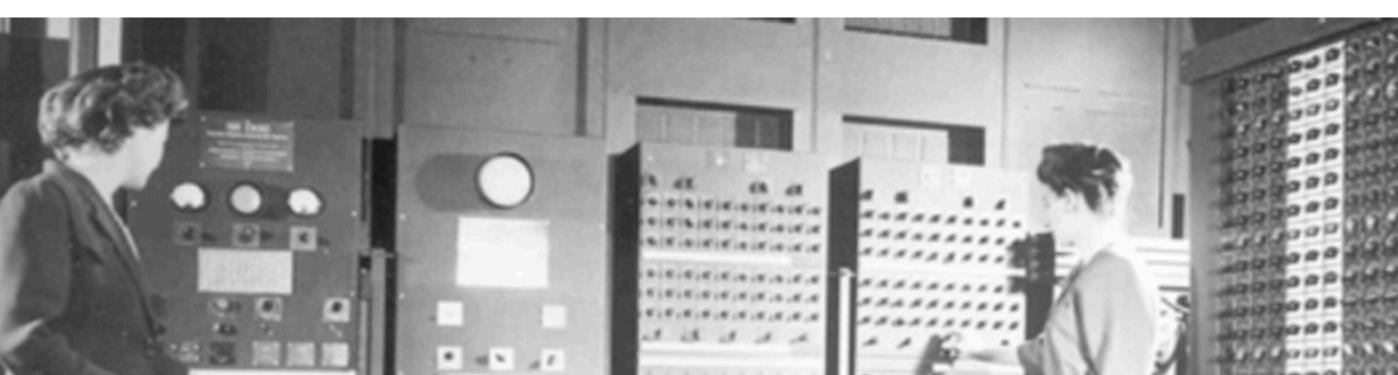
ENIACの女性たちと技術社会における女性の将来

この記事をシェア



近年「リケジョ」、「コード女子」など、理工系や情報工学系における女性の活躍に注目が集まりつつあるが、国内の大学の情報工学系学部における男女比は9:1程度にすぎない（文部科学省調べ、2006年）。しかしコンピューターの黎明期においては、むしろ女性の力によってプログラムが組まれていた。

この記事は、ESETが運営するマルウェアやセキュリティに関する情報サイト「We Live Security」の記事を基に、日本向けの解説を加えて編集したものである。



新着記事 NEW

2023.4.6

三菱UFJ信託銀行をかたるフィッシングについての注意喚起

2023.4.6

Webブラウザーのセキュリティ対策、適切にできていますか？

2023.4.5

関税等お支払いサイト(F-REGI 公金支払い)を装うフィッシングについての注意喚起

アクセスランキング



スマホがウイルスに感染！？不安に思ったら試したい5つの方法



そのウイルス警告は本物ですか？iPhoneが発する警告表示の意味とは？



iPhoneは盗聴されるのか？その確認方法はどう行うのか？

https://eset-info.canon-its.jp/malware_info/special/detail/160413_1.html

【連載】計算の歴史学とジェンダー誰が計算をしていたのか? (前山和喜)

B! [ツイート](#) [いいね!](#) 105 tumblr. + [LINE](#) noteで書く



[コーナートップへ](#)

【はじめに】

「コンピュータの歴史」、といわれて多くの人が思い浮かべるのは、機械の性能発展史やそれを開発した人々を英雄伝的に語る口調の物語ではないでしょうか。しかしこのような歴史の陰で、コンピュータに関わっていた無名の人々が数多く存在していたこともまた事実です。

本連載では日本計算史を専門とする研究者・前山和喜が、日本におけるコンピュータの黎明期から成熟期まで（1950年～1975年）左 計算機大体（右）

[カートを見る](#)

■サイト内記事検索

検索...



■本の検索 (版元ドットコム)

書名:

編著者名:

検索

★Google提供広告

★文学通信からのお知らせ



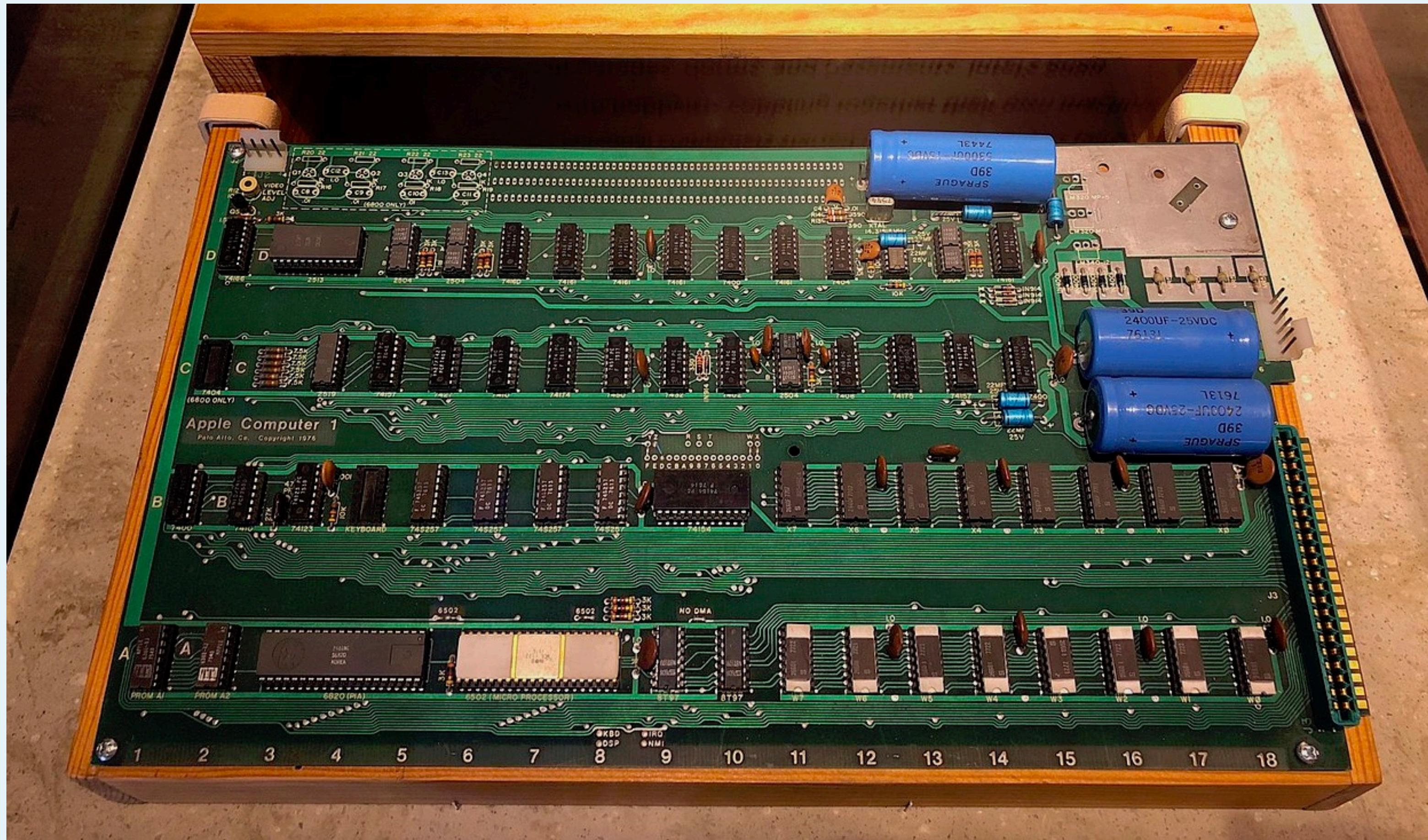
【連載】計算の歴史学とジェンダー誰が計算をしていたのか? (前山和喜,2020) 文学通信

<https://bungaku-report.com/computing.html>

Q : PCはいつ誕生しましたか？

1. 1950年代
2. 1960年代
3. 1970年代
4. 1980年代

Apple I(1976)



ArnoldReinhold, CC BY-SA 4.0, via Wikimedia Commons
https://commons.wikimedia.org/wiki/File:Apple_1_Woz_1976_at_CHM.agr_cropped.jpg

Xerox PARC 暫定版Dynabook (1972~1979)



VPRI Paper for Historical Context

Programming
Your Own
Computer

By Alan C. Kay

The computers of the 1980s will be easy to use and inexpensive, yet powerful enough to let us do things we have yet to imagine

Beth, a 14-year-old junior high student, lounged in the grass under a tree, enjoying the sunshine. She was watching pictures move about on the televisionlike display screen of a curious gadget she held in her lap. Her mother called from the house, "Beth, time to do your homework."

"I'm doing it, mother," Beth answered.

She began striking keys on the typewriterlike keyboard below the display screen. Part of a composition about space travel that she had begun the previous day appeared on the screen, along with a half-finished illustration and some references.

An awkward sentence caught her eye. She pointed a pencil-like stylus at the sentence, typed a revision, and watched the sentence change to her new phrasing. Then she finished the illustration with a drawing tool, filling in tones and drawing straight lines. When she made a mistake, she typed out instructions to undo what she had done, and then tried again.

Another of Beth's illustrations was a shuttle docked to a space station. She decided to animate it

183

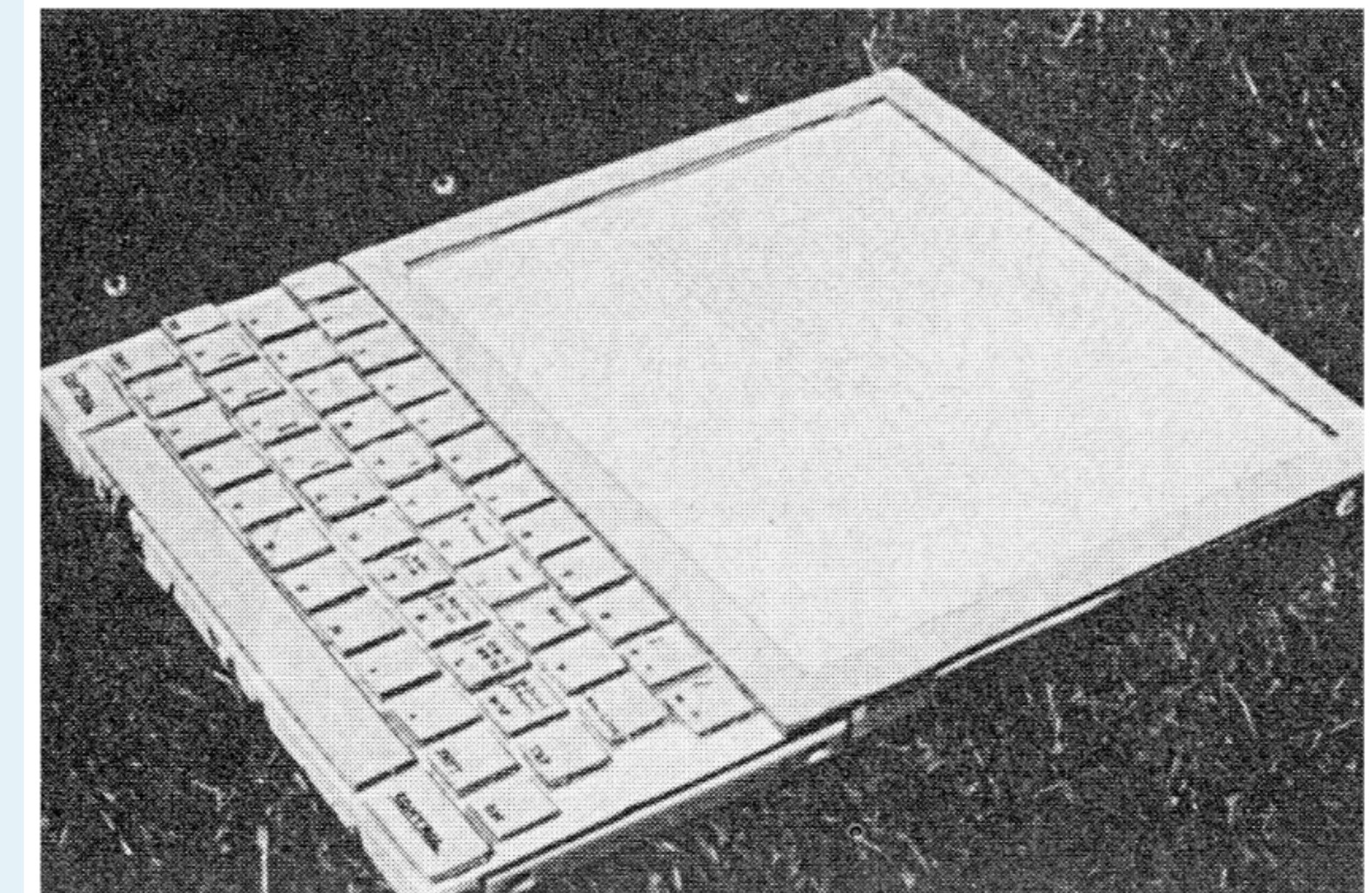


Figure 26.2. Mock-up of a future Dynabook.

Kay, Alan. 1979. "Programming Your Own Computer." In Science Year 1979. World Book Encyclopedia.

26. Personal Dynamic Media

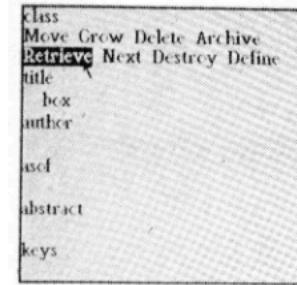


Figure 26.9. Retrieval in this filing tool is carried out by pointing to the command in the documents menu. The system will find every document with the title "box."

398

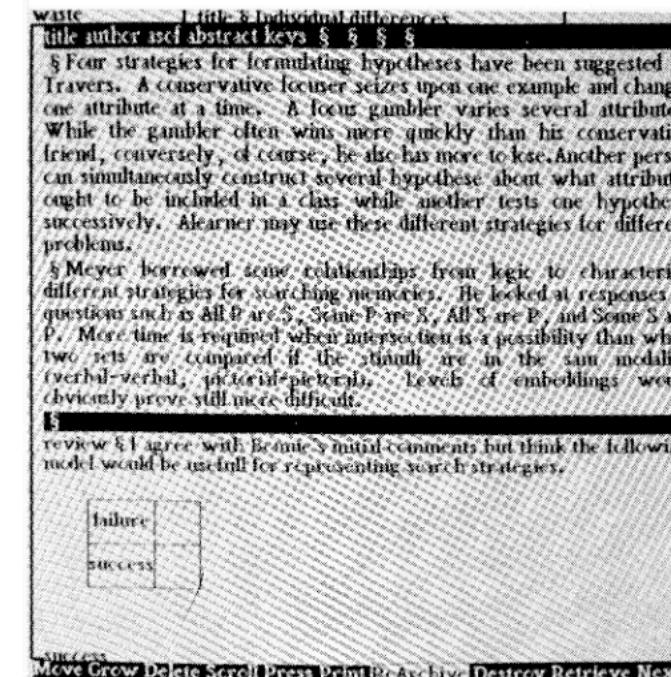


Figure 26.10. Here is a retrieved document that represents a description of a Smalltalk class definition.

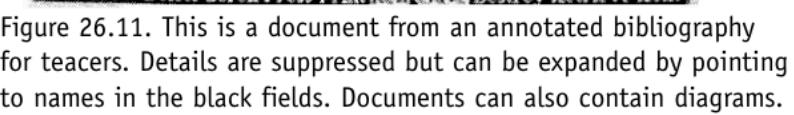


Figure 26.11. This is a document from an annotated bibliography for teachers. Details are suppressed but can be expanded by pointing to names in the black fields. Documents can also contain diagrams.

the NEWMEDIAREADER

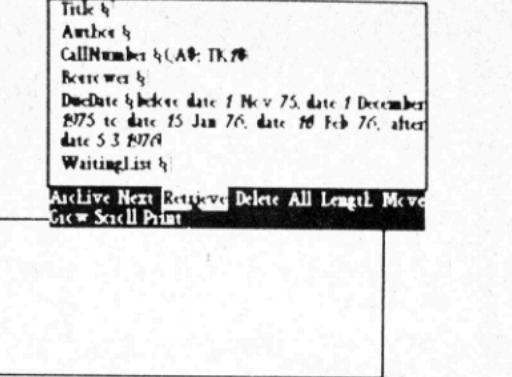


Figure 26.12. This retrieval request combines incomplete call numbers with date ranges. The example is taken from an experimental library system.

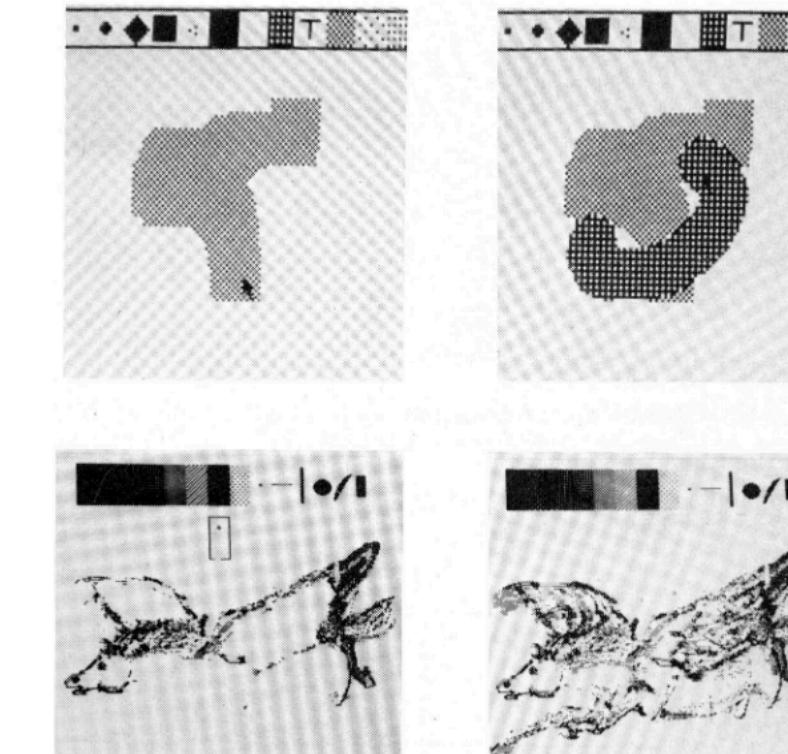


Figure 26.13. A sketch of Pegasus is shown being drawn with Smalltalk drawing tool. The first two pictures in the sequence show halftone "paint" being scrubbed on.

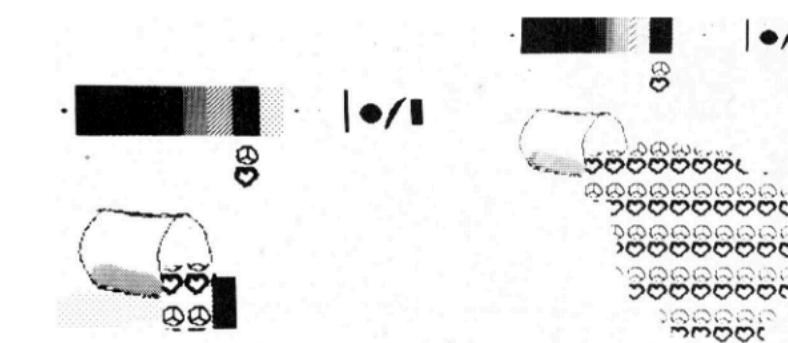


Figure 26.14. A sketch of a heart/peace symbol is created and used as a paint brush.

Macintosh (1984)

Electric Appliance (家電) からInformation Appliance (家情?) へ



Apple Museum (Prague) Macintosh 128k (1984), CC0

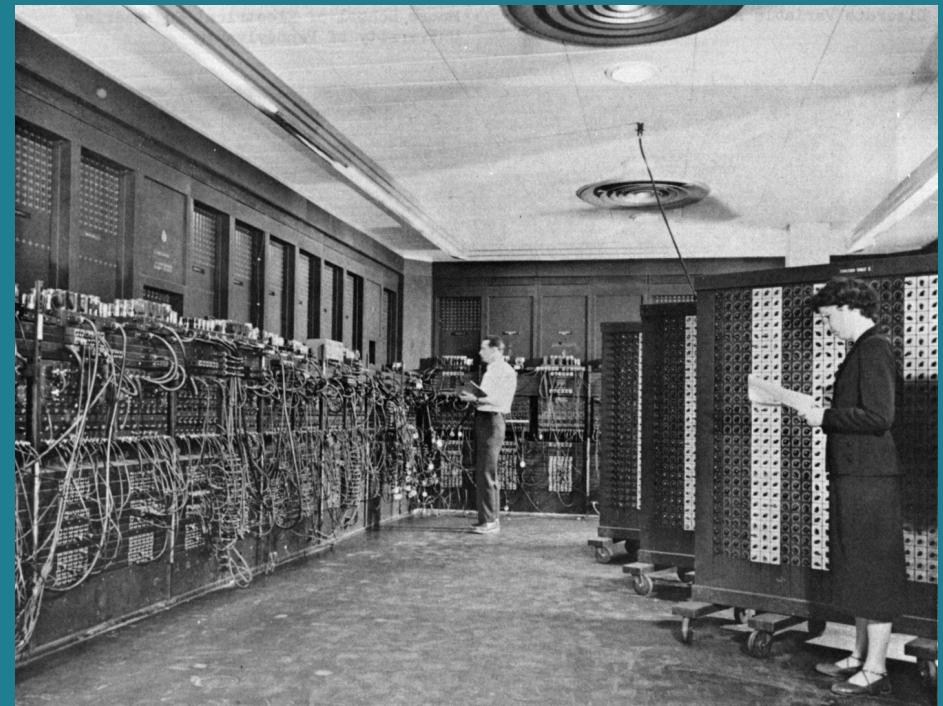
[https://ja.wikipedia.org/wiki/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:Apple_Museum_\(Prague\)_Macintosh_128K_\(1984\) - 1.jpg](https://ja.wikipedia.org/wiki/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:Apple_Museum_(Prague)_Macintosh_128K_(1984) - 1.jpg)



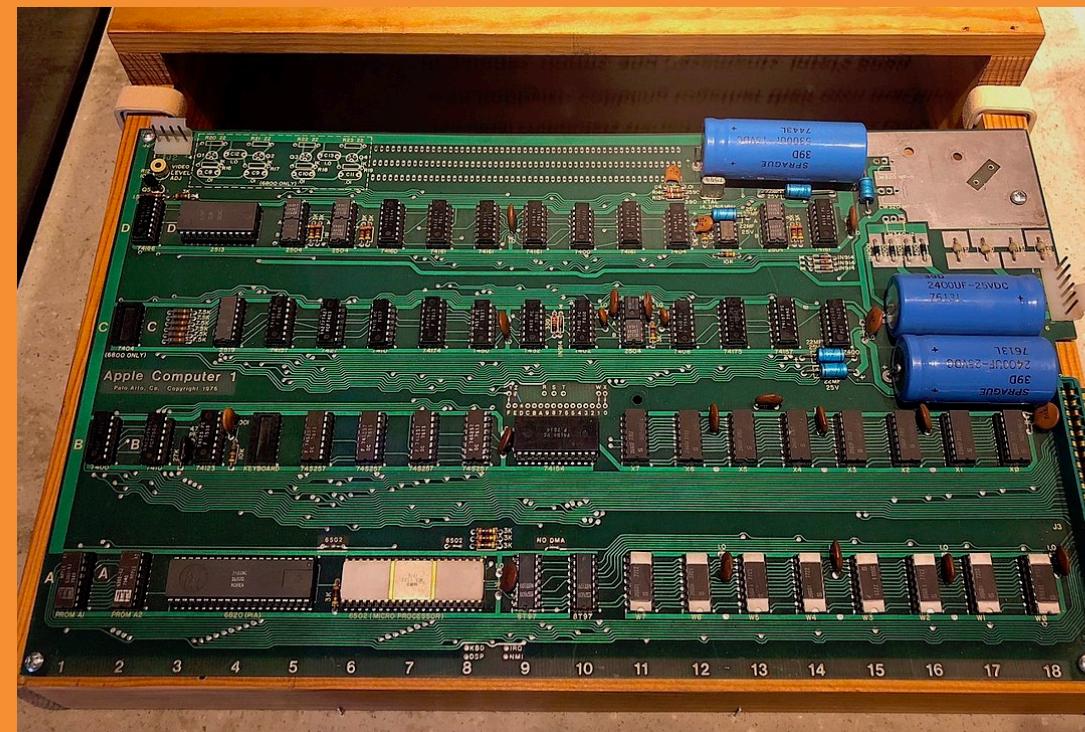
初代Macintoshの背面

(CC-BY-SA, <http://www.allaboutapple.com/>)

コンピューターの使用=
プログラミング
(In Lab)



ギークのためのプロダクト



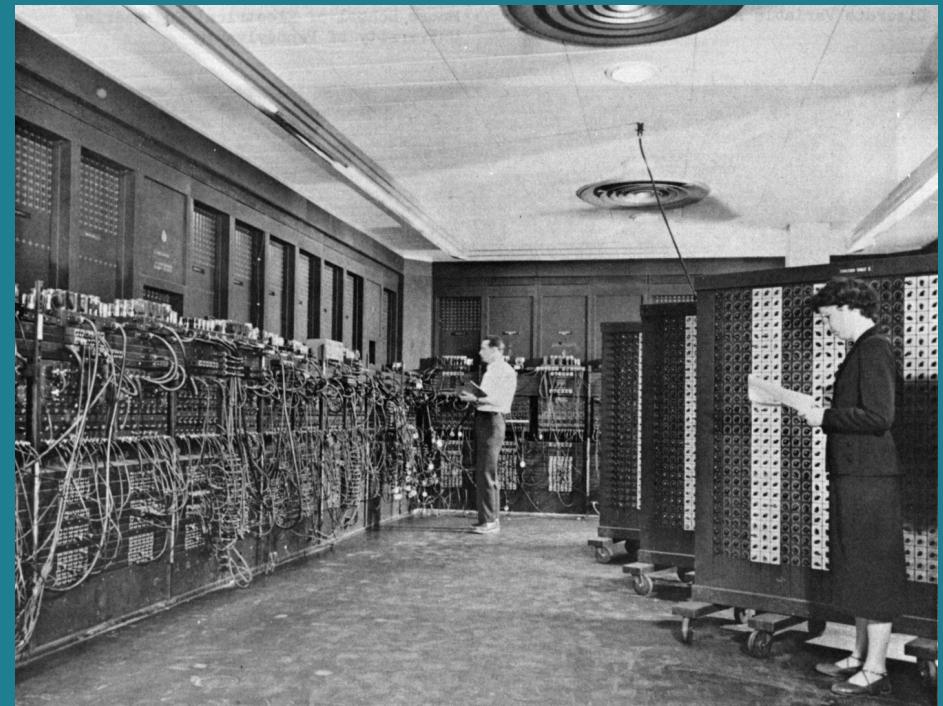
プログラムを用いて個人の能
力、表現を拡張



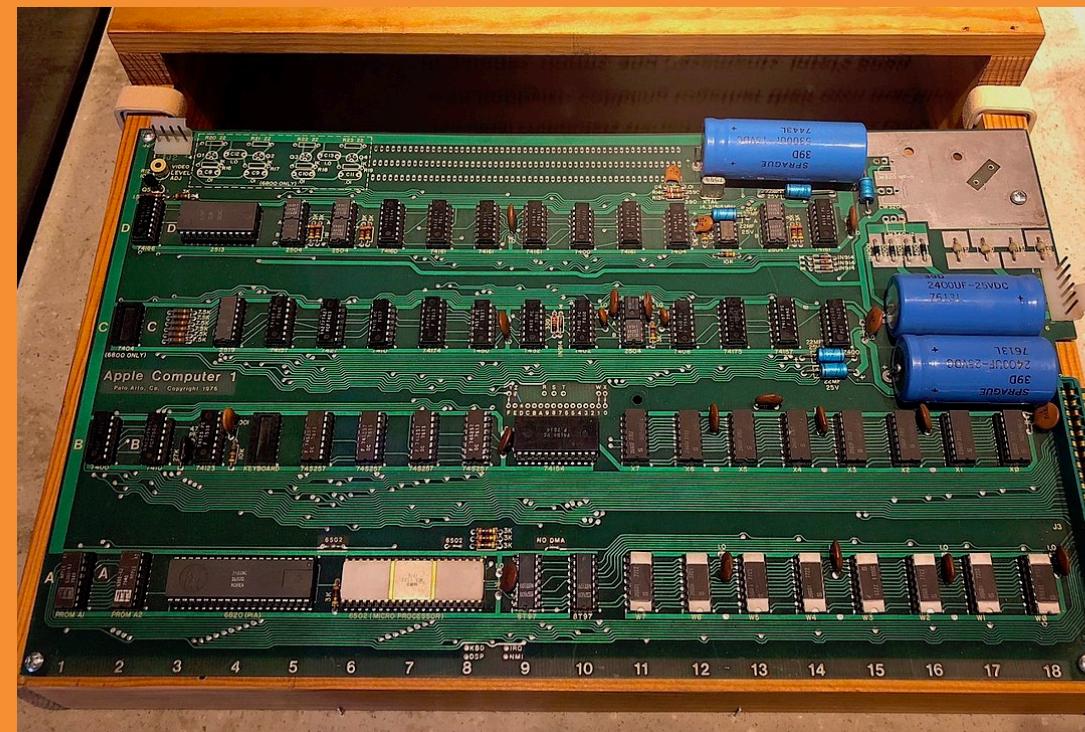
プログラムを知らなくても簡単
にカスタマイズ



コンピューターの使用=
プログラミング
(In Lab)



ギークのためのプロダクト



プログラムを用いて個人の能
力、表現を拡張



プログラムを知らなくても簡単
にカスタマイズ

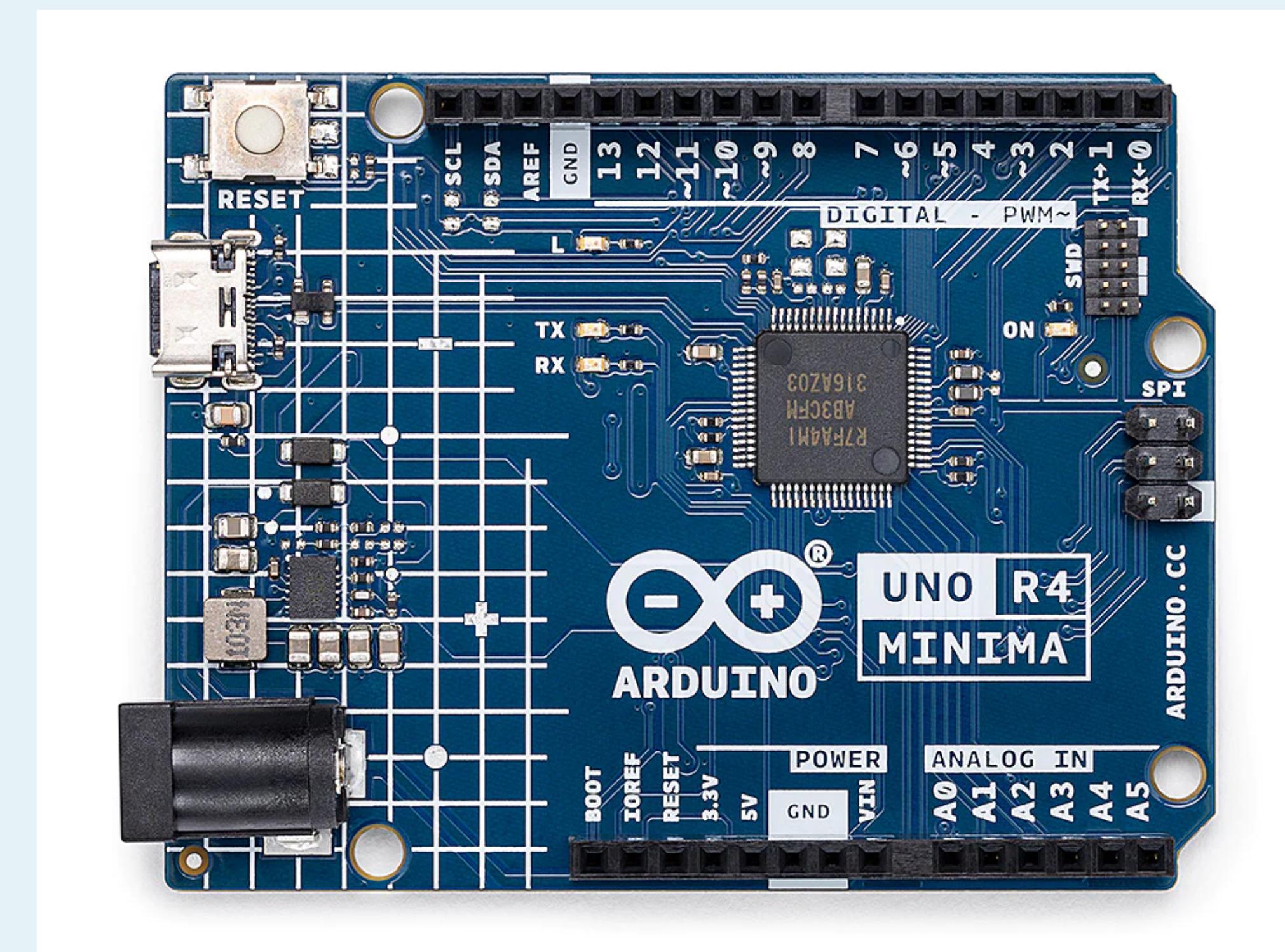
Customize



プログラミング

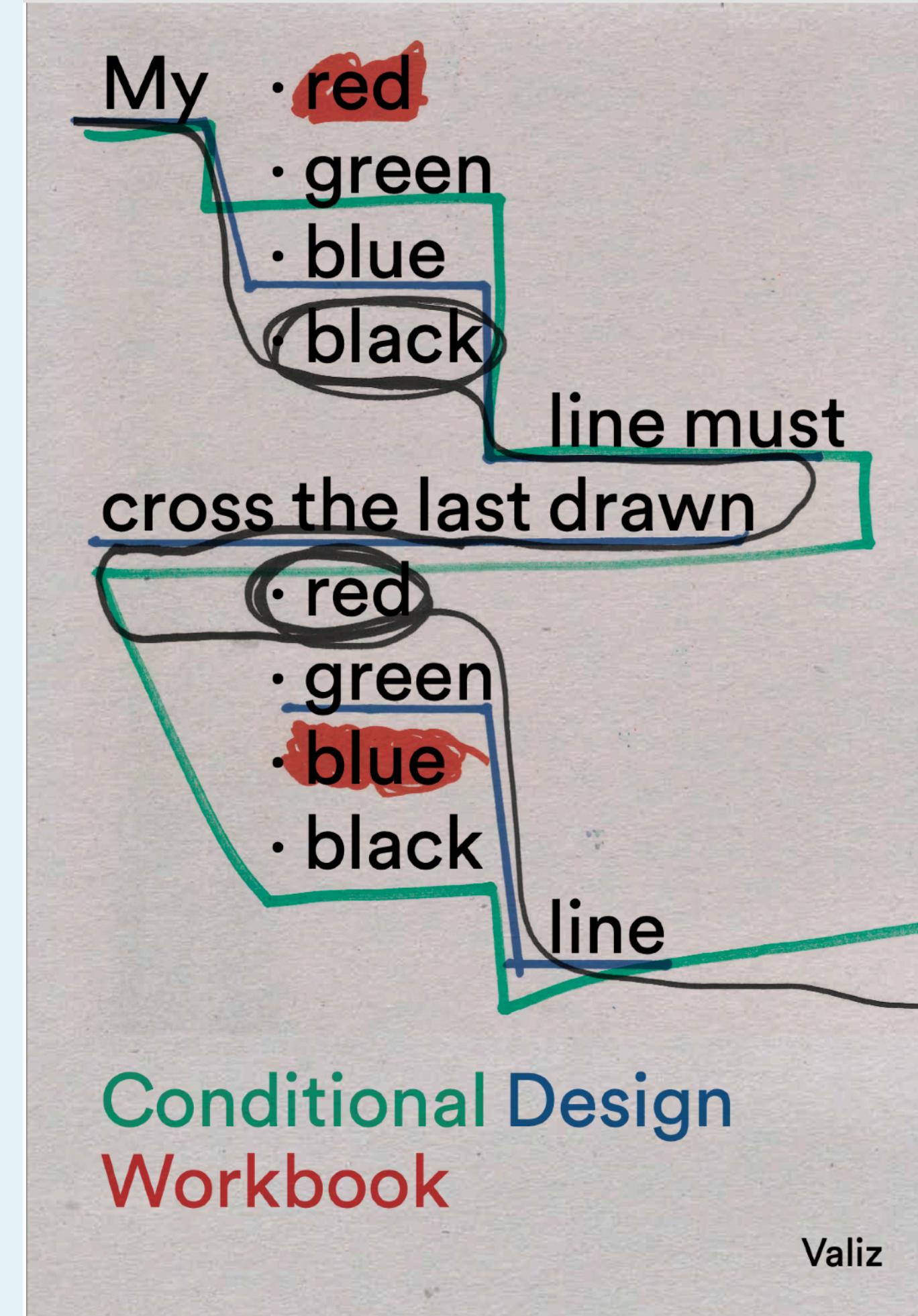
≡

コンピューターを
自分だけの道具にする手段



WS：コンピューターなしでの
プログラミング

Conditional Design

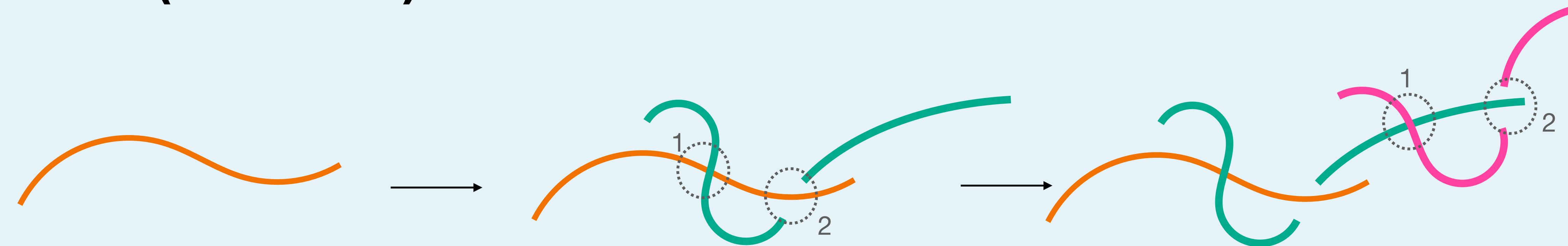


Luna Maurer, Edo Paulus, Jonathan Puckey and Roel Wouters(2008) <https://www.conditionaldesign.org/>
<https://www.valiz.nl/en/publications/conditional-design-workbook>

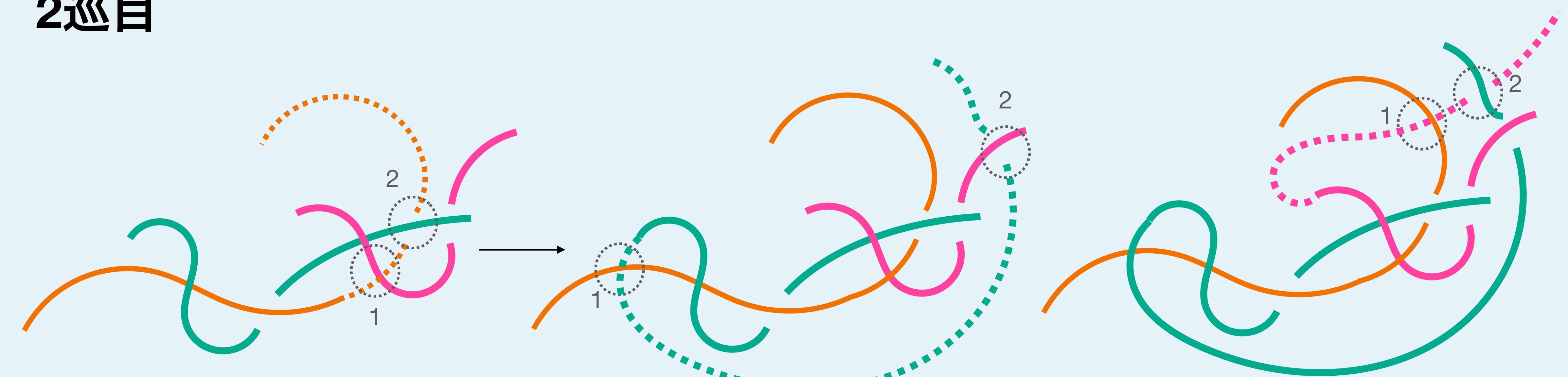
やってみよう

- 4人違う色のペンを持って、順番を決める。
- はじめに、一人目は適当に線を書く。一巡目は、順番に前の人との線を跨いだあと、くぐるように線を書く。
- 順番に、自分の線をそれぞれ延長する。その際、他の人の線を跨いだあと、別の色をくぐるようにする。
- 書ける場所がなくなったら、別の場所から線をスタート。

1巡目(3人の場合)



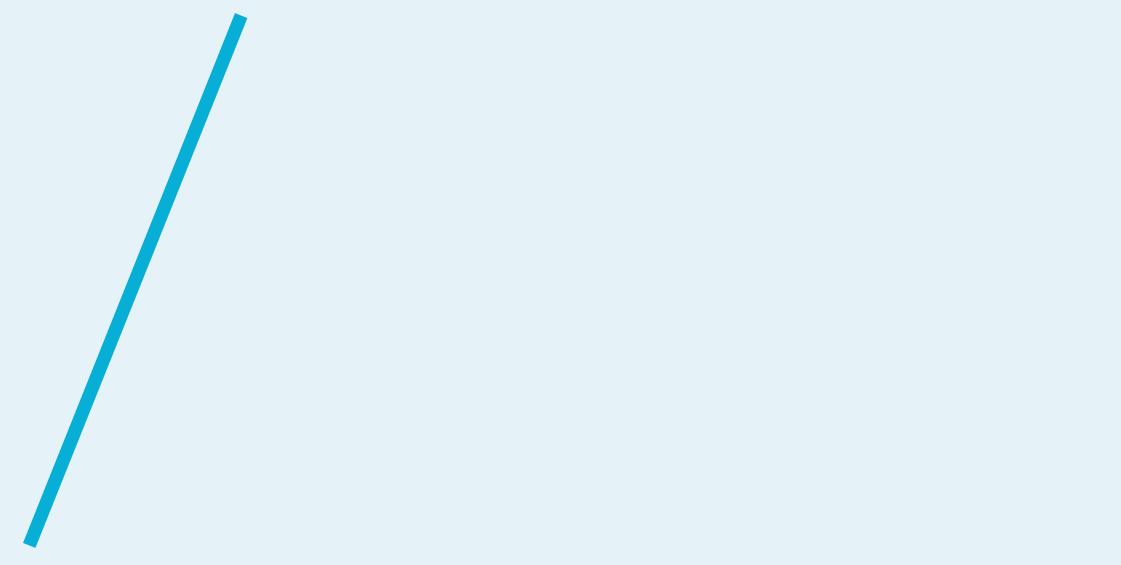
2巡目



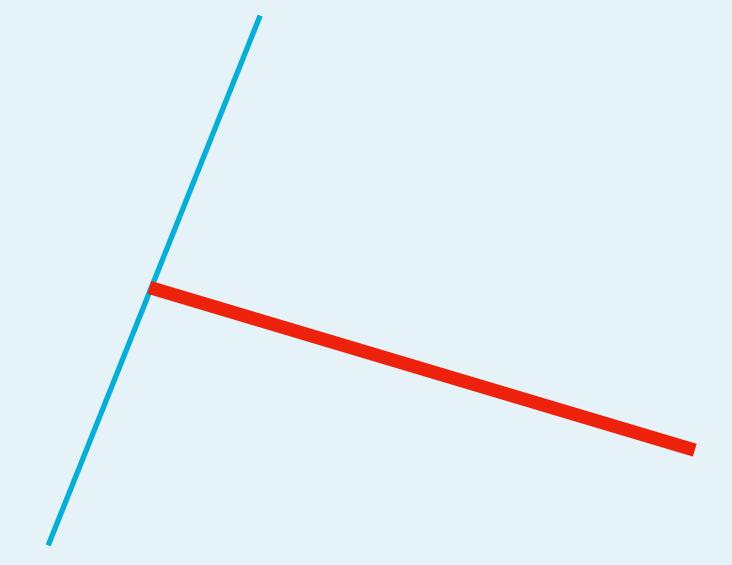
やってみよう2

- 協力して「コードとデザイン」というテキストを書く。
- 時計回りに順番に1人ずつ直線を引く。
- 奇数番目のは、既存の線の中央から垂直に好きな長さの線を引く。
- 偶数番目のは、いずれかの線の端同士を繋ぐ直線を引く。

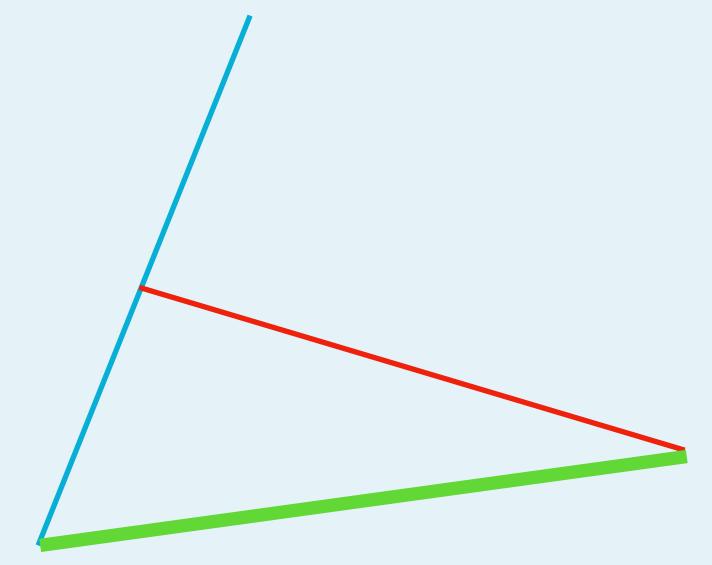
スタート



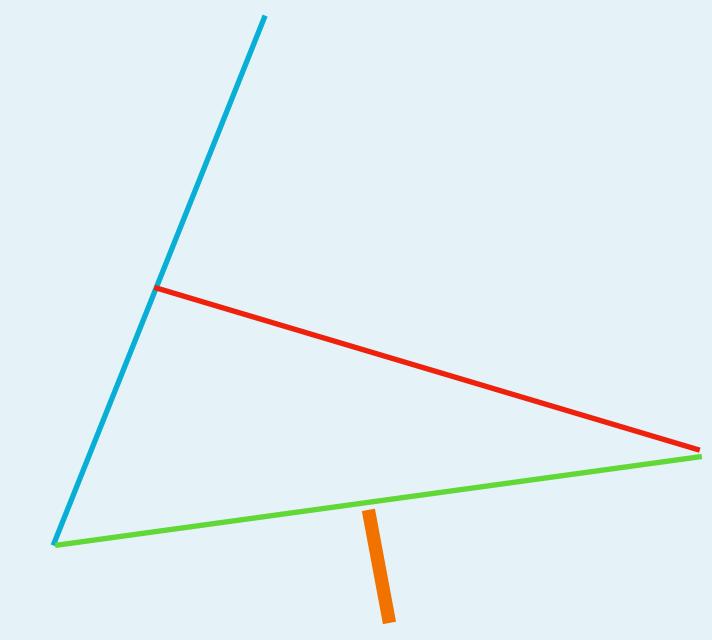
奇数ルール

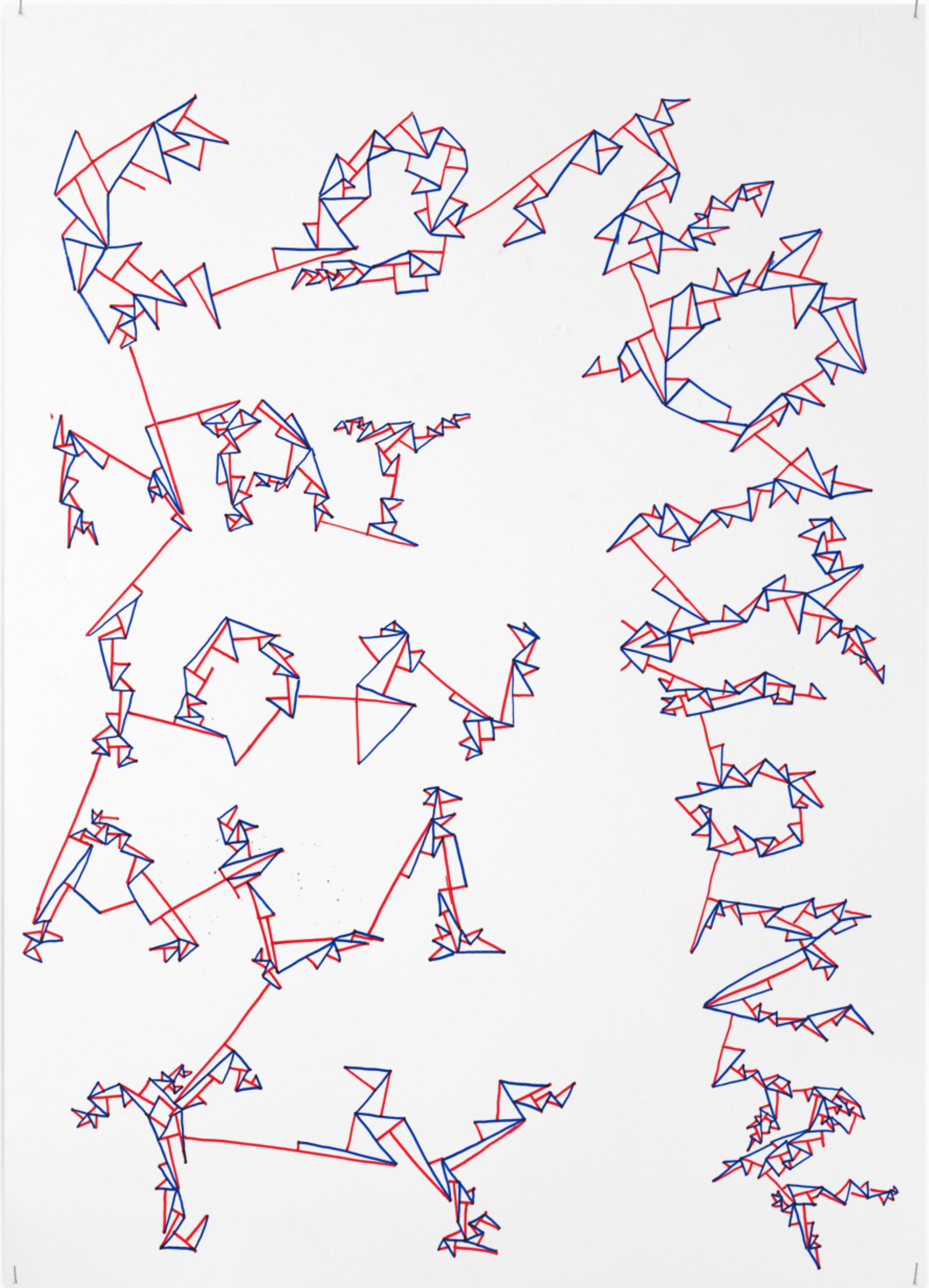


偶数ルール



奇数ルール





Hong Kong & Dutch Designers Cross-cultural Poster Exhibition

[https://conditionaldesign.org/workshops/
conditional-nationality/](https://conditionaldesign.org/workshops/conditional-nationality/)

単純なルールからの複雑な創発

ロジックを定める → ロジックに収められないものを認識する



Image Creator のしくみを確認しますか? [驚くようなおすすめのプロンプト] を選択し、[作成] を選択します。



13



お任せで探す

作成は手順の 1 つです

作成する数が多いほど、より適切に作成できます。
画面の指示に従って操作します。わかりやすい単語
を含めます。そして、思い描くものが手に入るまで繰
り返します。

ガソリン スタンドの赤いスポー



ネオン、未来的なスタイル



鉛筆画



暗くて不吉



日没までドライブ

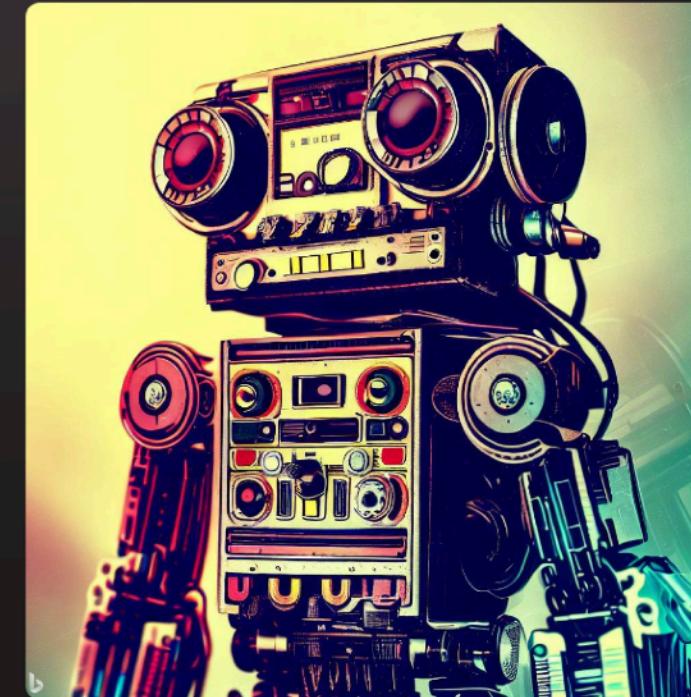
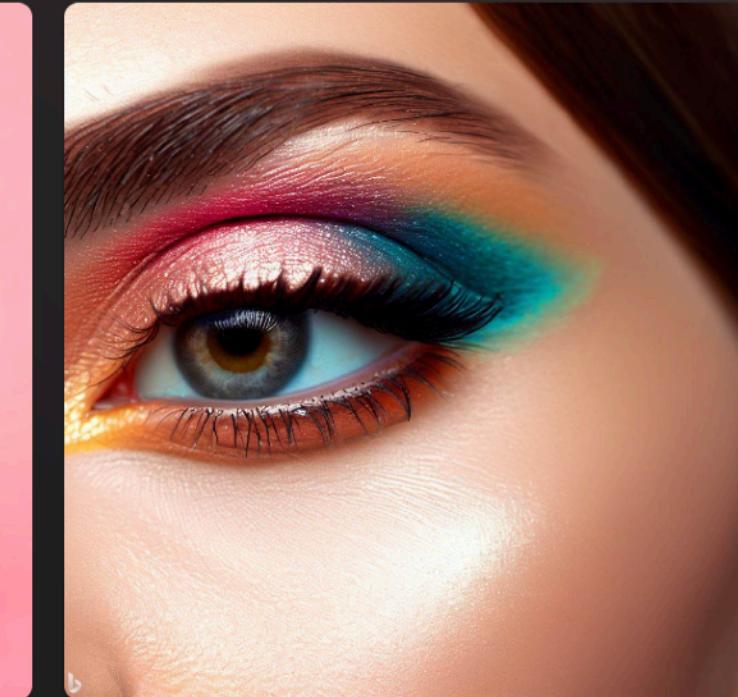
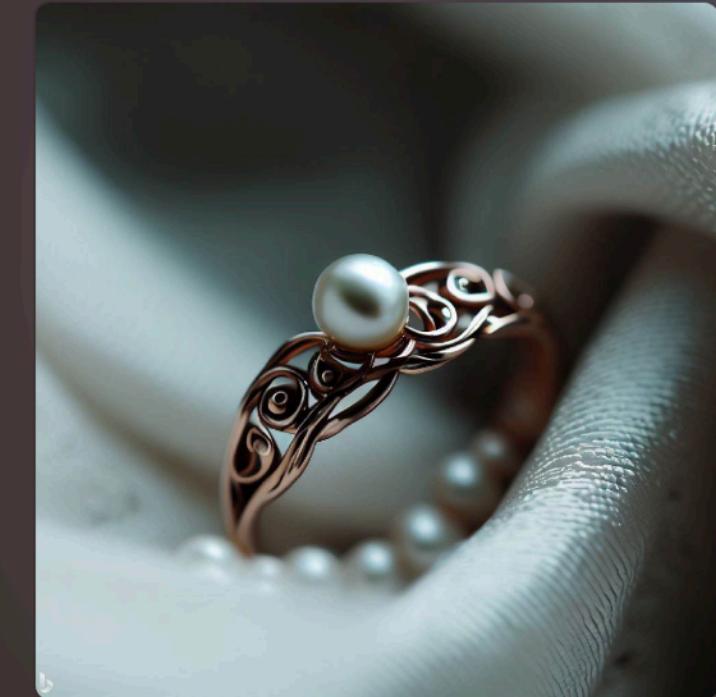
アイデアを探す

作品

AI で作成



ヘルプ



プライバシーと Cookie

コンテンツ ポリシー

使用条件

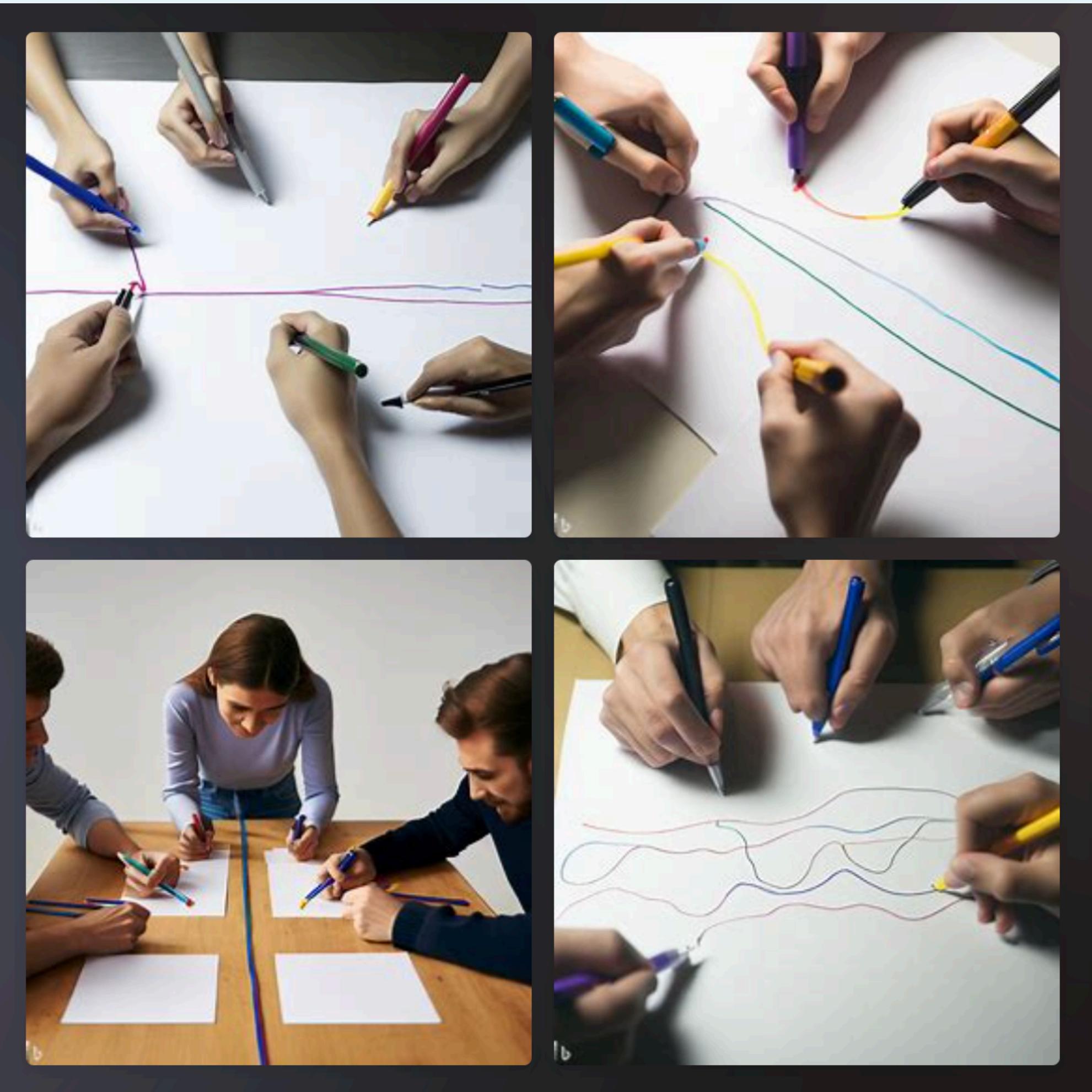
フィードバック

AI で作成

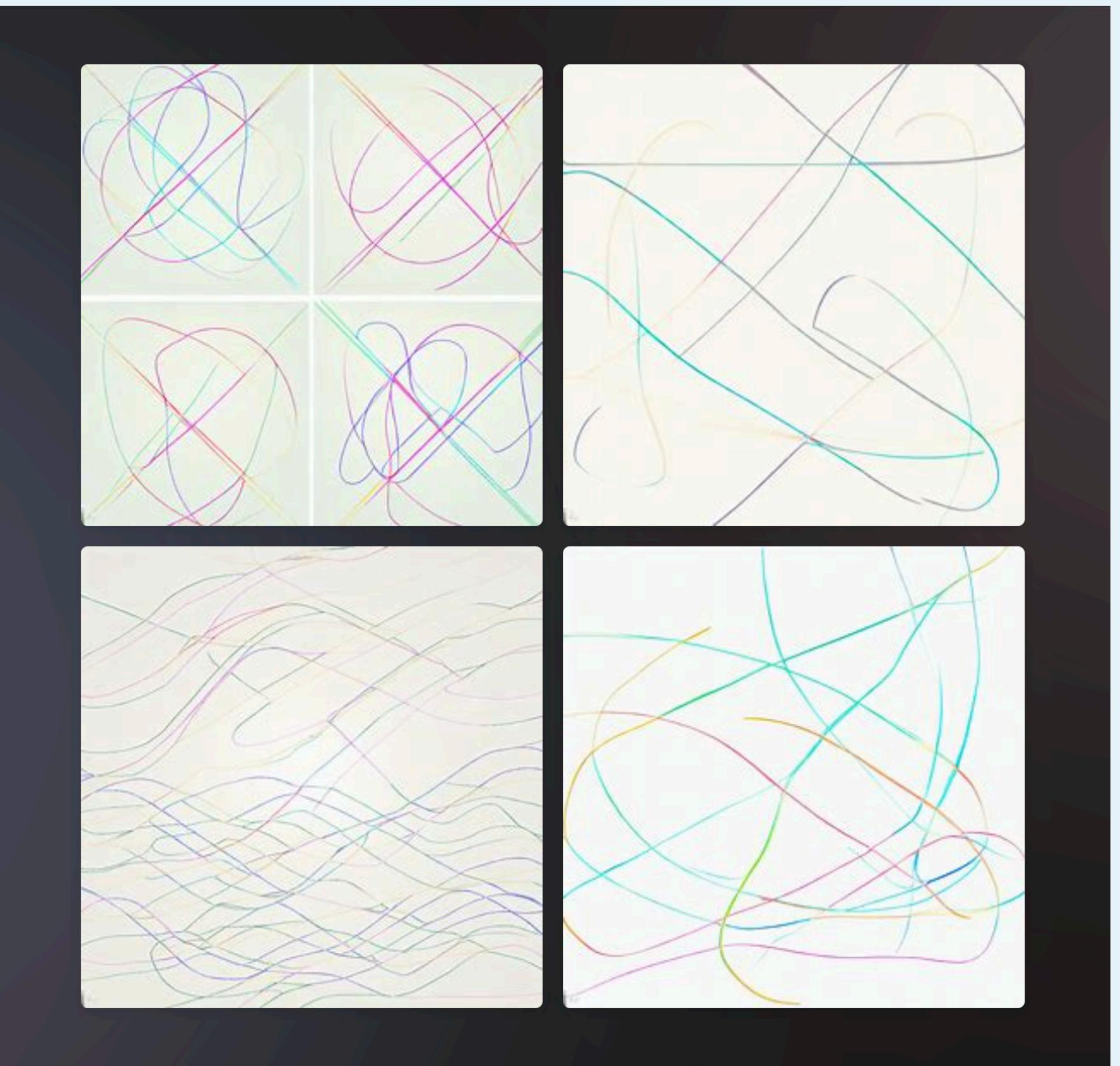
フィードバック

<https://www.bing.com/images/create>

- Simulate a situation where four people with pens make drawings on paper.
- Each of the four people has a different color pen and draws a line according to the following rules
 - 1. draw the longest possible line over a period of 20 minutes.
 - 2. keep the pen off the paper during that time.
 - 3. the pen may be held still for a maximum of 5 seconds.
 - 4. each line must never cross each other.

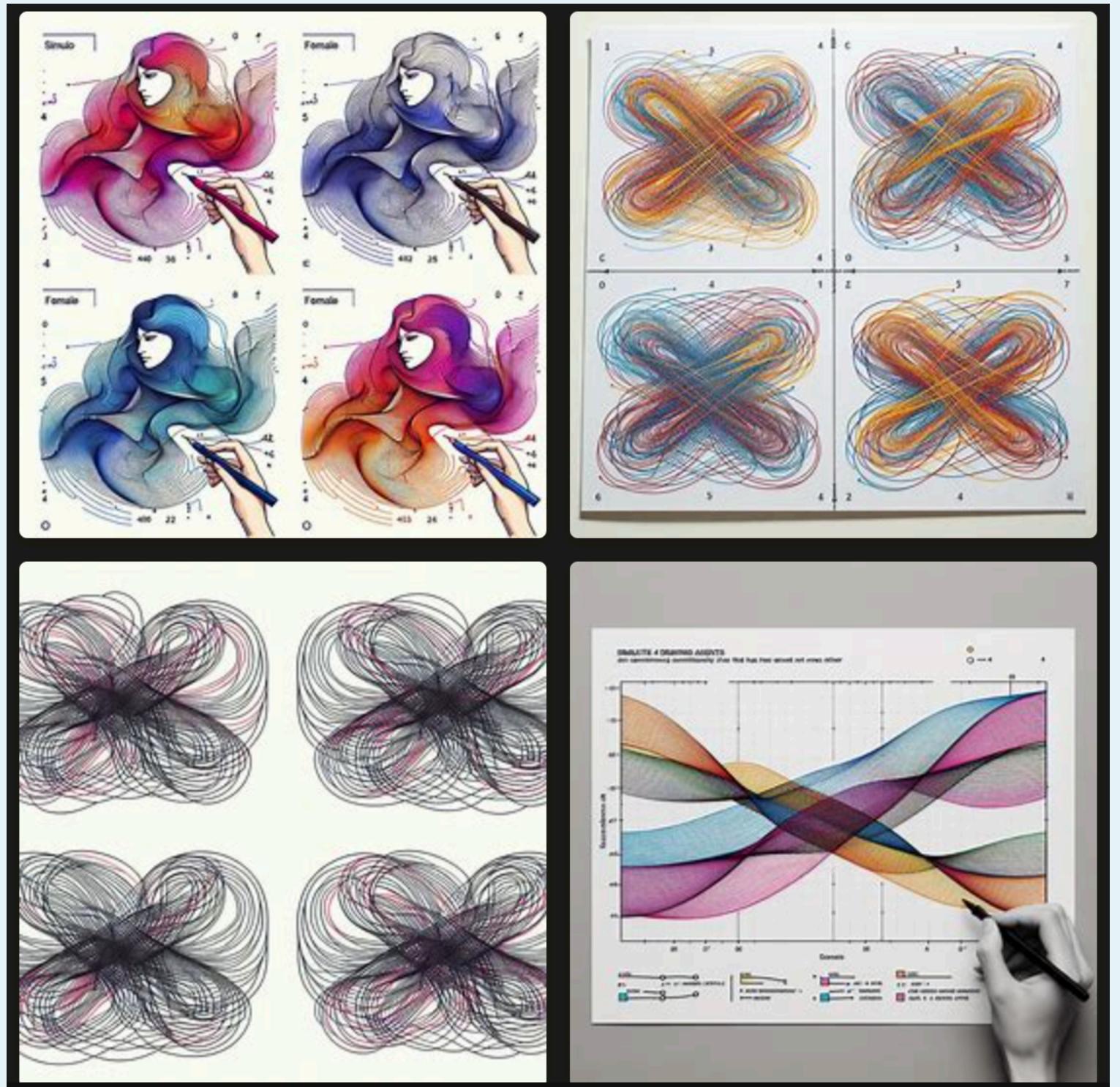


- Simulate 4 drawing agents that has a unique color. Each agents draw a constant and random lines continuously and simultaneously on a plain white paper. Each lines must not cross each other. Each agents try to draw a long line as possible.



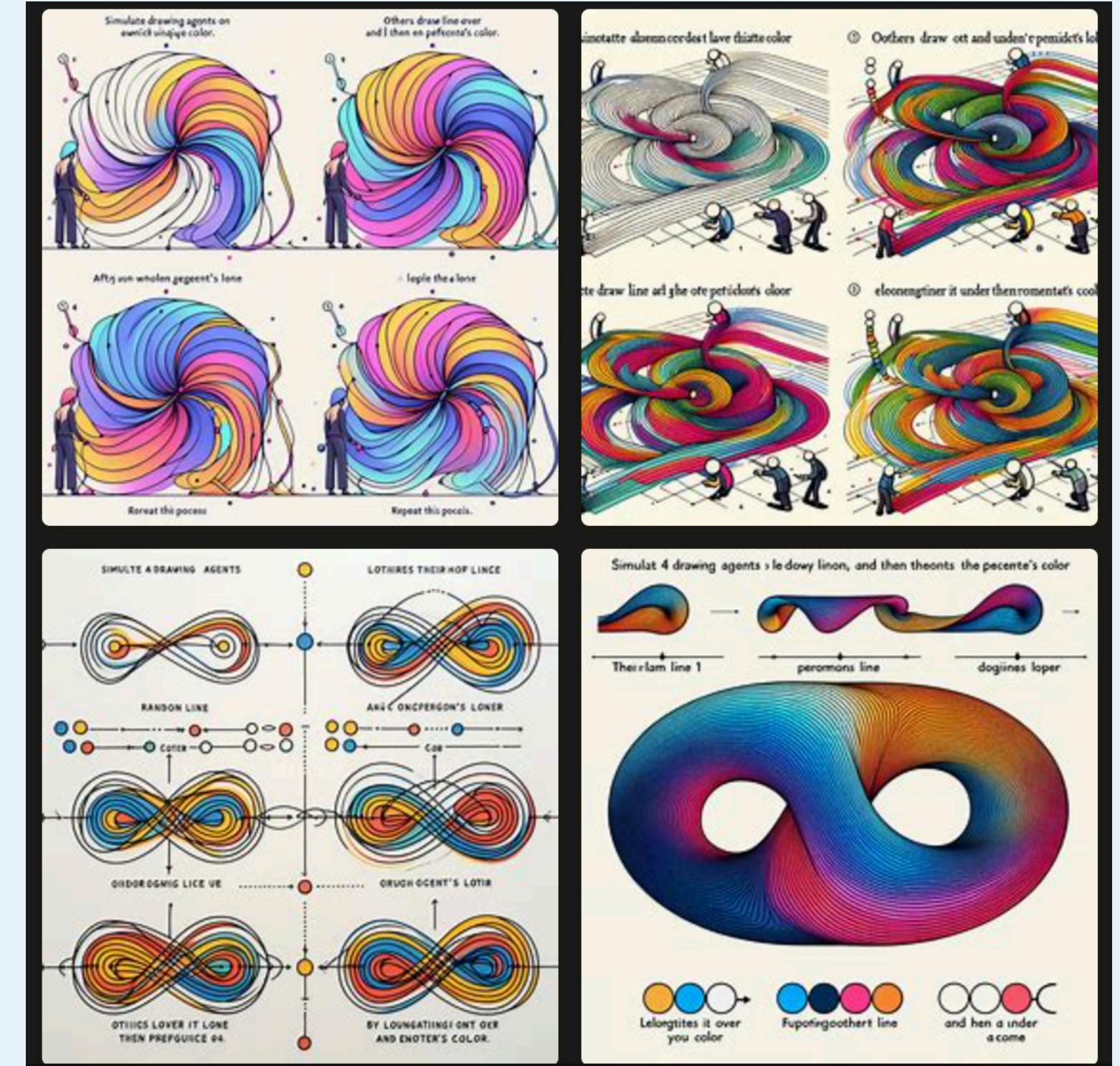
2023年に実行したとき

- Simulate 4 drawing agents that has a unique color. Each agents draw a constant and random lines continuously and simultaneously on a plain white paper. Each lines must not cross each other. Each agents try to draw a long line as possible.



2024年に実行したとき

- Simulate 4 drawing agents on the white paper that have unique colors. First, agent 1 draws a random curved line. Others draw line by looping it over and then under a line of precedent's color. After that, each agent elongates their own line by looping it over the other agent's line and then under the even another agent's line. Repeat this process.



履修上の注意点

スケジュール1

*シラバスから変更あり

- 1. (4/12) イントロダクション 電気を使わないコンピューティング
- 2. (4/19) 電子工作脱入門 (Victorian Oscillator)
- 3. (4/26) Paper Circuitと論理回路
- 4. (5/10) 二進数ゲーム (Binary Card Game)
- 5. (5/17) Handmade Computer
- 6. (5/24) ソフトウェアとアルゴリズム—Arduinoを使ってみよう
- 7. (5/31)秋葉原遠足

スケジュール2

- 8. (6/7) 入力を考える (雑なマウス)
- 9. (6/14) 入力を考える2 - ProcessingとArduinoの連携、オブジェクト指向プログラミング (一人用PONG)
- 10. (6/21) 出力を考える - Arduinoでの音声プログラミング
- 11. (6/28) 制作：最終課題構想①
- 12. (7/5) デジタルファブリケーション：入力機器（可変抵抗）を作る
- 13. (7/12) 制作：最終課題構想②
- 14. (7/19) 課題発表展示インストール
- 15. (7/25) (木曜振替) 展示撤収・発表ふりかえりとまとめ

授業（後も含めた）目標：

あなたにとっての「パーソナルコンピューター」を作れるようになること

目標&最終課題

あなたにとっての「パーソナルコンピューター」を作れるようになること

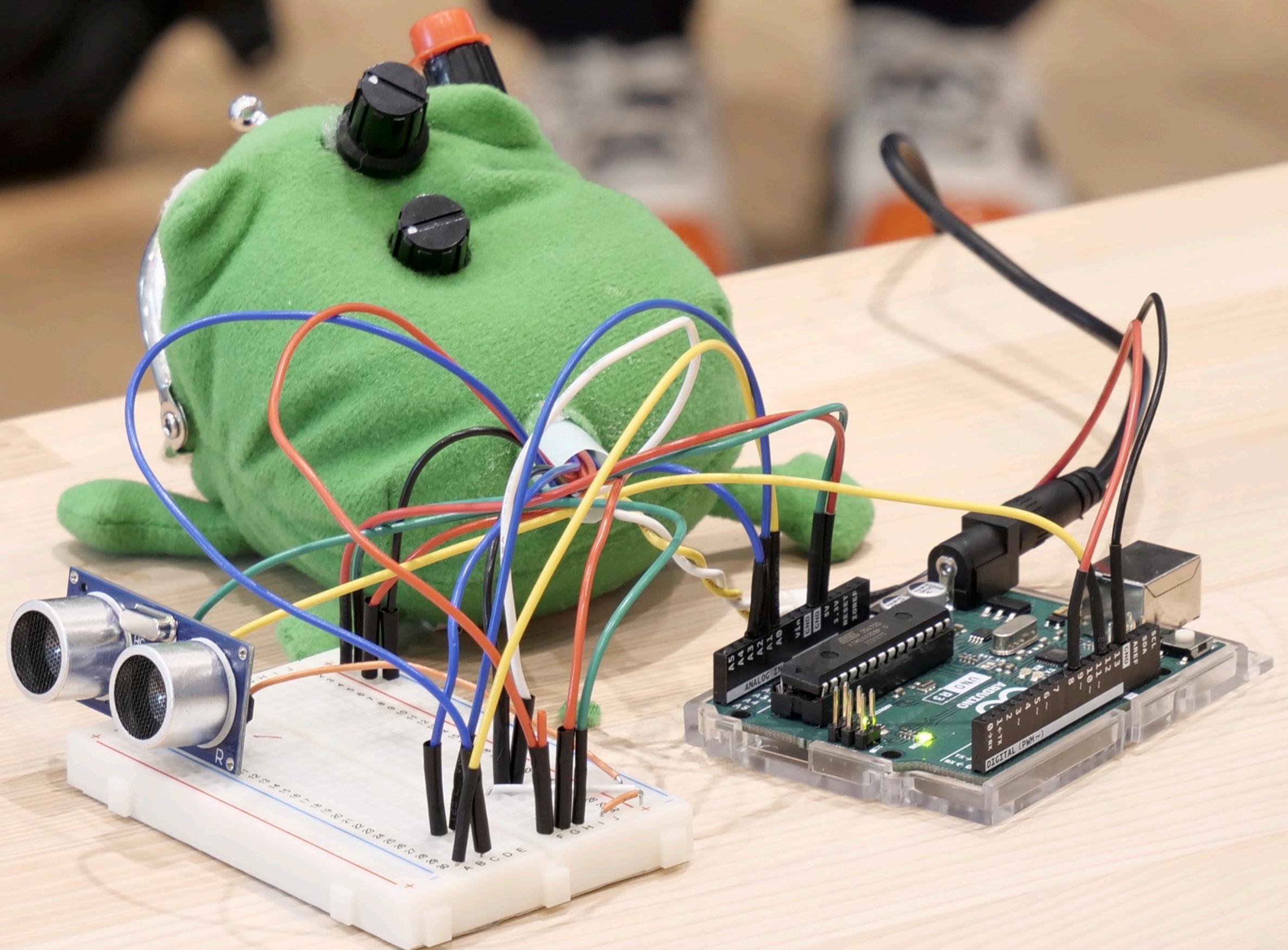
- 例えば：自動ドローイングマシン、アクセサリー、入力装置
- Or、授業内の小課題を自分なりに発展させる
 - 例えば、Conditional Designの新しいルールを考えるとかもあり◎
- 自由枠：自分の課題制作等に授業内で学んだ内容を反映する
- 技術的な難易度よりも、なぜそれを作ったのか伝わるように
- 進捗が無くても（無い人ほど）授業に来る（重要）

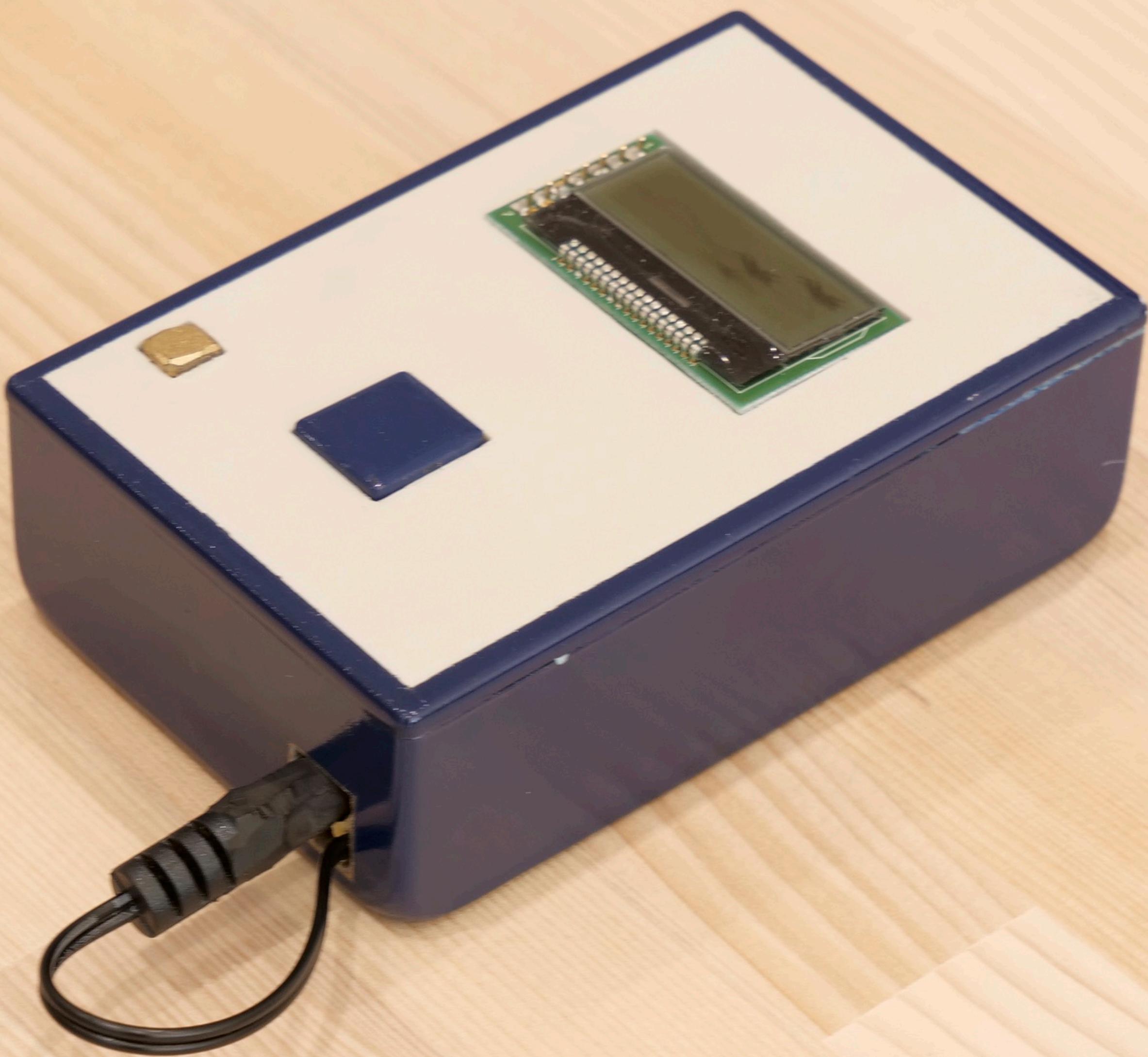




the MIDI Controller

controller







成績評価

- ・ 成績は出席点を30%、授業中に数回提示される小課題点を30%、最終課題点を40%として評価する。
- ・ 出席点は毎回授業の最後にGoogle Formでの感想、質問シートの提出をもってカウントする。必ず毎回その回でわからなかったことや気になっている内容について最低1つ質問してください！次の授業でフォローアップします。
- ・ あらゆる課題について、無条件で、未提出は0点、期限を過ぎての提出は減点する。
- ・ **課題を含めた授業関連のやりとりには全てClassroomを利用します！！**

機材関連

- AMC演習室のコンピューターのデータはシャットダウンすると削除され、設定もリセットされます。
- データを保管しておきたい人は、ネットワーク上の“shared”の中のコードとデザインのフォルダに、自分の名前のフォルダを作成して保存してください。
- ただし、他の人からも見える&編集できてしまうので、USBメモリ等を持参するのをお勧めします

- Google Formで履修アンケートを提出した中から抽選を行います。
- Classroomで履修アンケートを配布するので、回答できたら今日の授業は終了

Image Credits

- Clément Hélardot on Unsplash(<https://unsplash.com/ja/%E5%86%99%E7%9C%9F/95YRwf6CNw8>)
- Alienware on Unsplash(<https://unsplash.com/ja/%E5%86%99%E7%9C%9F/Bp3KmTZQIfw>)
- Daniel Korpai on Unsplash(<https://unsplash.com/ja/%E5%86%99%E7%9C%9F/HyTVmlxKVOA>)
- Harrison Broadbent on Unsplash(https://unsplash.com/ja/%E5%86%99%E7%9C%9F/fZB51omnY_Y)
-