

バイブルコーディング入門

弘前実業高校 情報処理科 2年生向け 特別授業

2025/10/22

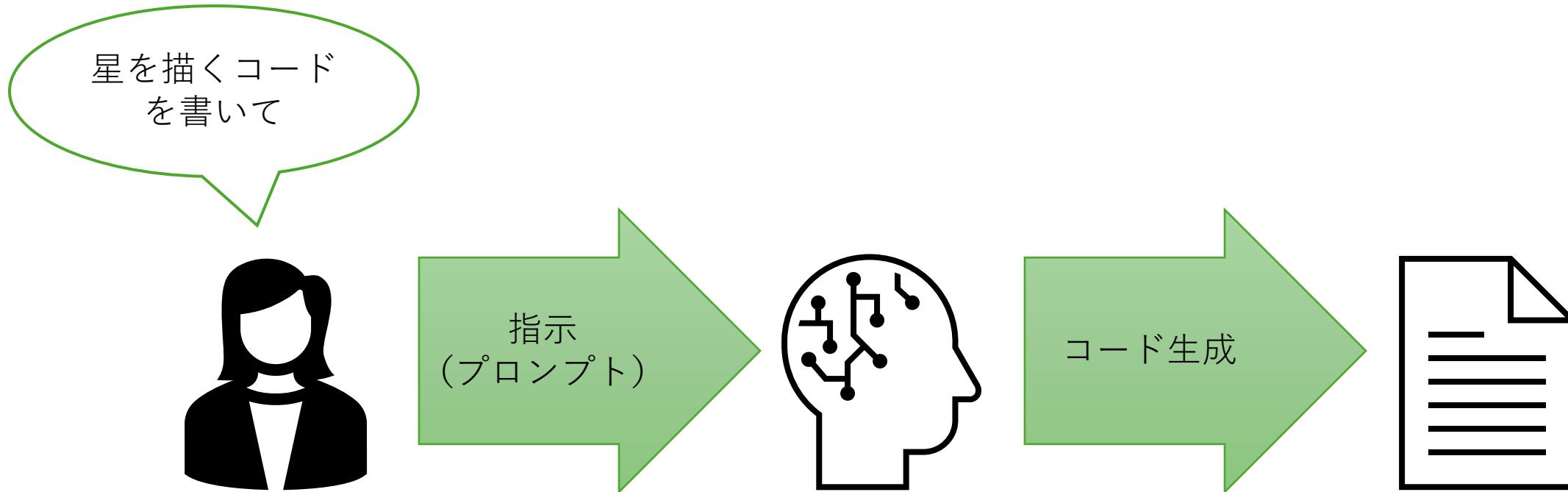
バイブルコーディングとは

コードを書かないコーディング

バイブルコーディング

vibe (バイブル)
ノリ・雰囲気・フィーリング
(音楽用語)

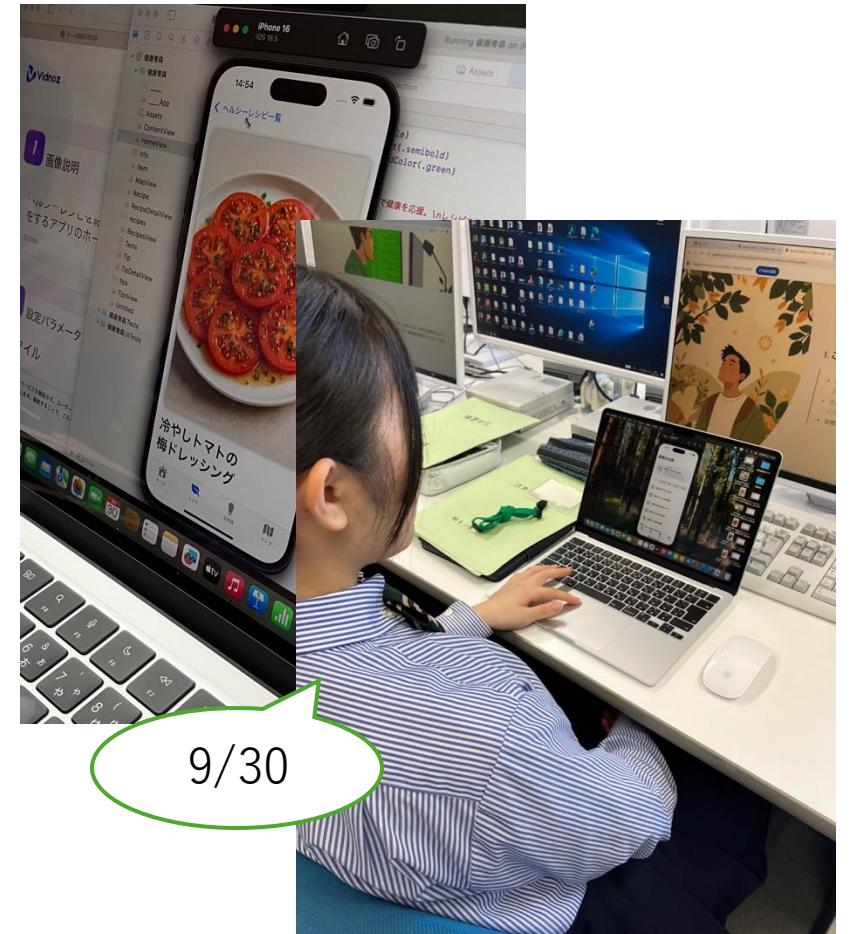
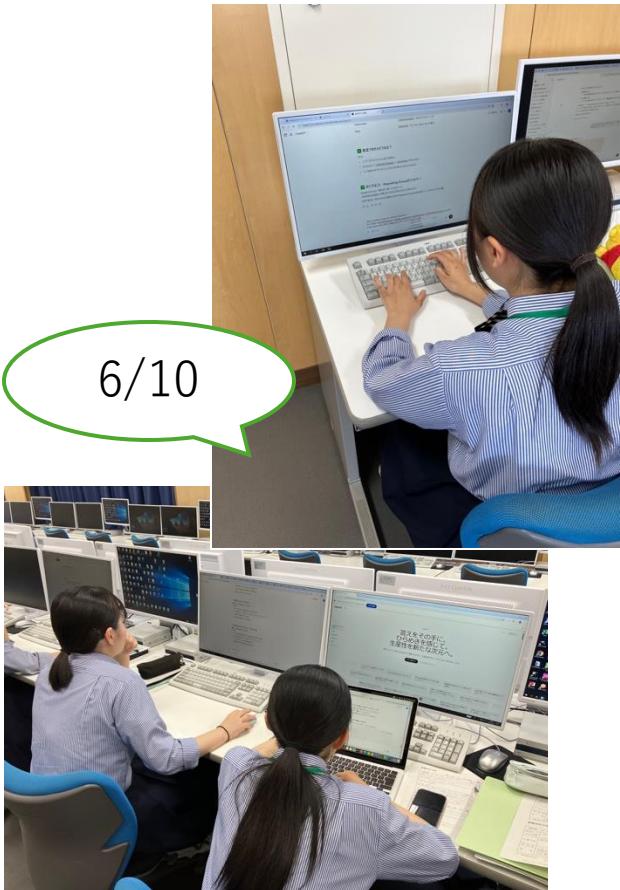
- コーディングを生成AIにまかせる開発手法



プログラミング未経験・知識なしでもプログラムが作れる？

事例紹介（現在進行形）

- 課題研究でバイブコーディング@青森商業高校



実践！バイブルコーディング

Webアプリを作ろう

Webアプリを作ろう！

- いろいろなWebアプリ

メール

YouTube 動画視聴

Microsoft 365 Copilot オフィスアプリ

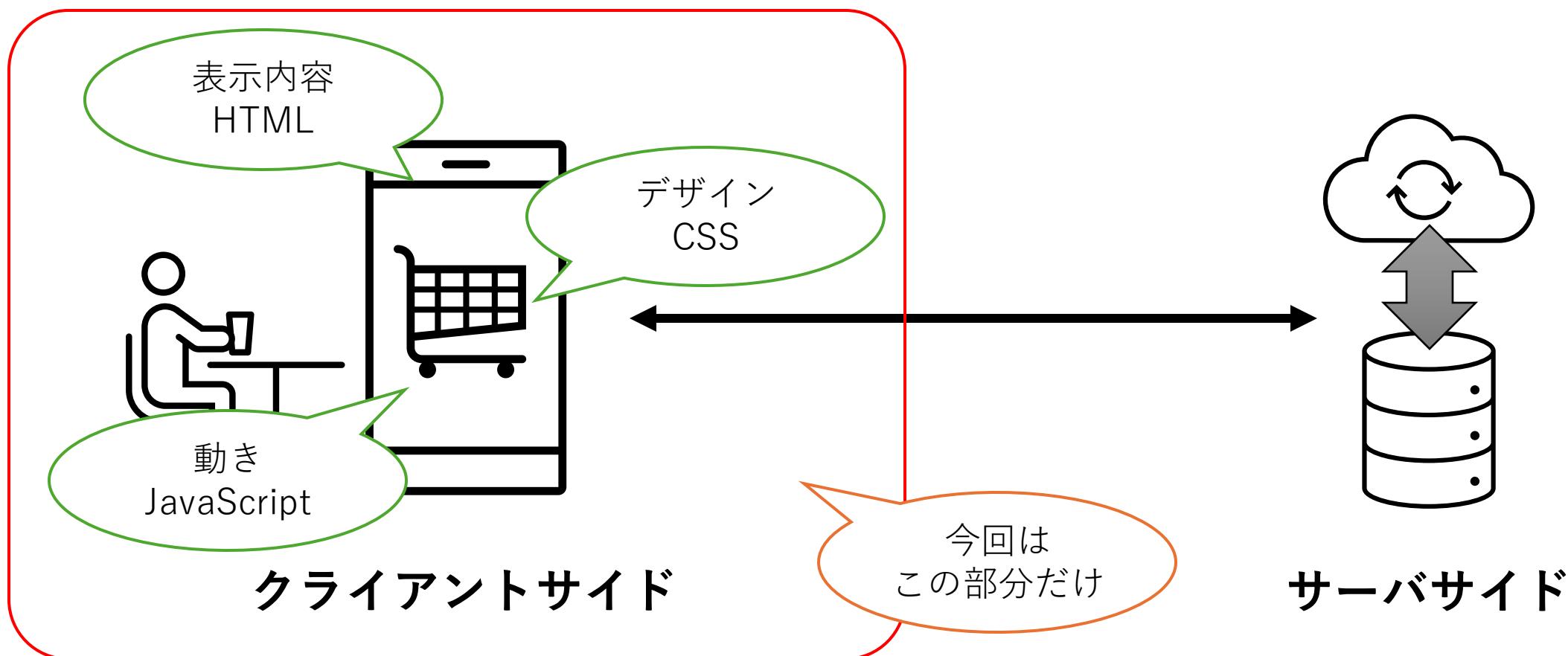
Minecraft ブラウザゲーム

The collage consists of four main images:

- Gmail (メール):** Shows the inbox screen with several emails from Roger Nelson, Alan..Shirley, Keith, Lauren, Jason Coleman, Edward Wang, and Gloria Hill. A green oval highlights the "Compose" button.
- YouTube 動画視聴 (動画視聴):** Shows the YouTube homepage with various recommended videos like "BUYER'S GUIDE", "LAUDA KAMPF", and "POLAND". A green oval highlights the search bar.
- Microsoft 365 Copilot オフィスアプリ (オフィスアプリ):** Shows the Microsoft 365 Copilot interface with sections for Outlook, Word, Excel, PowerPoint, OneNote, OneDrive, and Teams. A green oval highlights the "Microsoft 365 Copilot で使用" section.
- Minecraft ブラウザゲーム (ブラウザゲーム):** Shows a screenshot of the Minecraft Java & Bedrock edition game environment with various characters and blocks.

Webアプリのしくみ

- HTML + CSS + JavaScript で動くアプリ



コードマップ

- 夜空のシミュレータを作る

- ① ヘッダにボタン1個、メインは空
- ② ボタンを押すとコンソールにログ
- ③ ログをメイン画面にも表示
- ④ 画面分割 + Canvasで星を1つ表示
- ⑤ ボタンを2つに(星の数を増やす/減らす)
- ⑥ ボタンを3つに(アニメーションのオン/オフ)

- 進め方

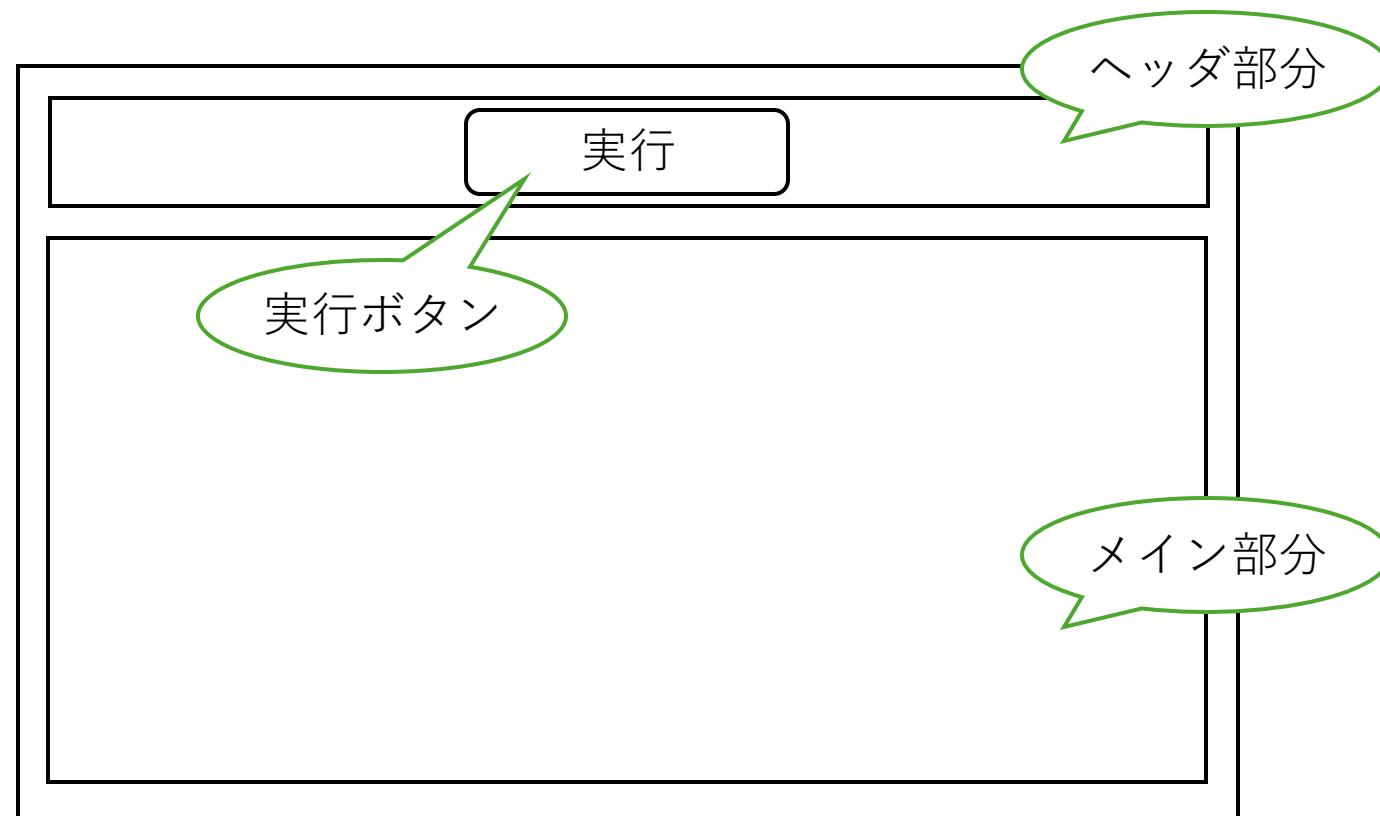
- 生成AIを使ってHTML+CSS+JavaScriptのコード生成
- ブラウザで開いて動作確認

バイブルコーディングのコツ

- 上手なプロンプト（コツ）
 - シンプルなものから少しづつ機能を追加する
 - 条件は毎回書く
 - 「Done条件」を示す（自己チェックしてくれる）
- うまくいかない／違うものができた
 - 前に戻ってやり直す ← 状態にナンバリングしておく
 - 最初からやり直す ← プロンプト（指示）を記録しておく

Step① ヘッダにボタン1個、メインは空

- ・画面にボタンを表示する。



Step① ヘッダにボタン1個、メインは空

- ・プロンプトの例

html+css+jsで動くブラウザアプリのコードを生成してください。
条件として、1ファイル(stars.html)で動き、外部ライブラリは使わず、できるだけシンプルで、
わかりやすい日本語コメントを入れてください。
stars.htmlを上書きできる完全なコードとして出力してください。

共通指示

(1) ヘッダにボタン1個、メインは空

動作内容: ブラウザで開くと上部ヘッダに「実行」ボタンが1つ表示される。メイン領域は空。

Done条件: ボタン表示のみ。押しても何も起きない。コンソール表示もなし。

個別指示

Step② ボタンを押すとコンソールにログ

- ・コンソールにログ表示



Step② ボタンを押すとコンソールにログ

- ・プロンプトの例

(1)を改造します。

条件として、1ファイル(stars.html)で動き、外部ライブラリは使わず、できるだけシンプルで、わかりやすい日本語コメントを入れてください。
stars.htmlを上書きできる完全なコードとして出力してください。

共通指示

(2) ボタンを押すとコンソールにログ

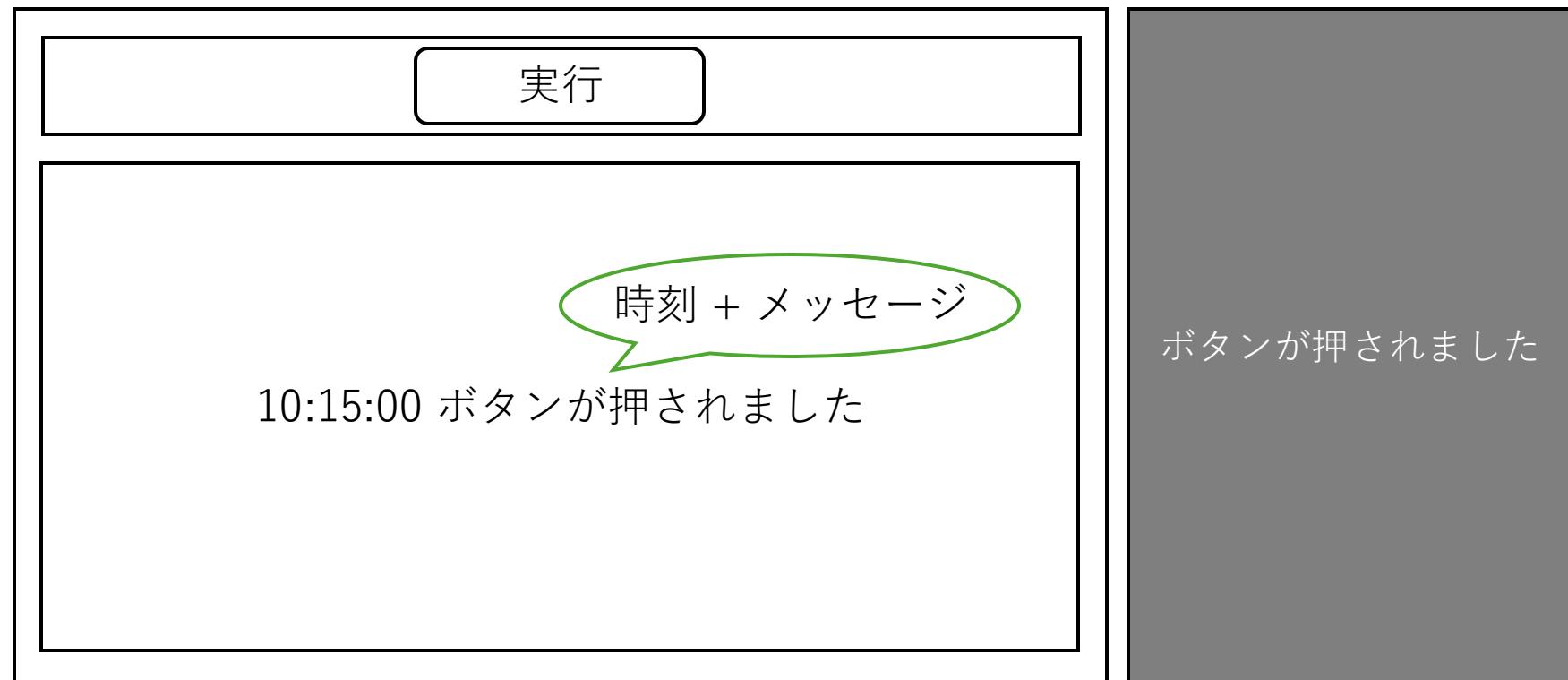
動作内容: ボタンを押したらブラウザのコンソールに「ボタンが押されました」と出力する。

Done条件: コンソールにメッセージが出る。

個別指示

Step③ ログをメイン画面にも表示

- 画面にもログ表示



Step③ ログをメイン画面にも表示

- ・プロンプトの例

(2)を改造します。

条件として、1ファイル(stars.html)で動き、外部ライブラリは使わず、できるだけシンプルで、わかりやすい日本語コメントを入れてください。
stars.htmlを上書きできる完全なコードとして出力してください。

共通指示

(3) ログをメイン画面にも表示

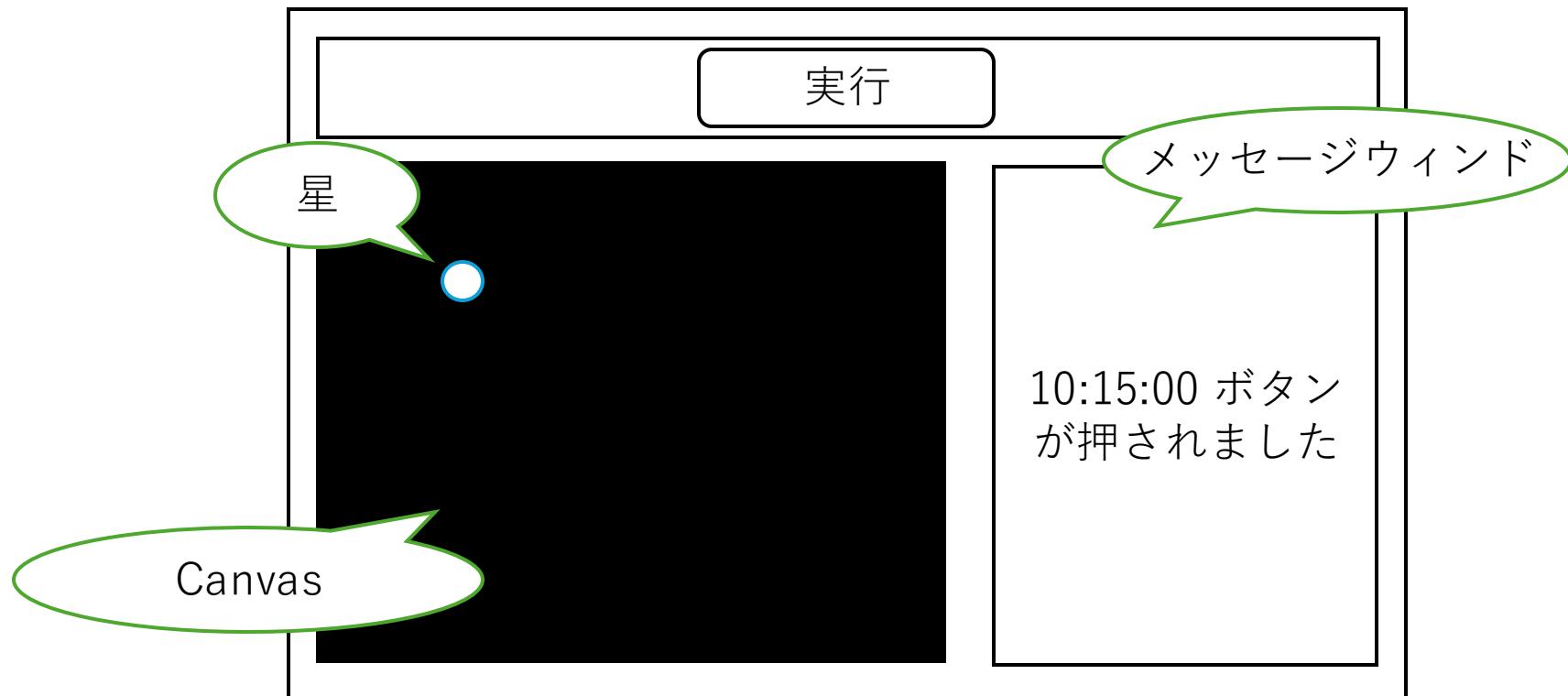
動作内容: ヘッダのボタンはそのままで、メイン領域にログ表示欄(#log)を作り、ボタンを押すたびに先頭に追記される形で、時刻付きでメッセージを表示する。コンソール出力は残したままにする。

Done条件: コンソールと画面の両方にログが出る。

個別指示

Step④ 画面分割 + Canvasで星を1つ表示

- ・画面にもログ表示



Step④ 画面分割 + Canvasで星を1つ表示

- プロンプトの例

(3)を改造します。

条件として、1ファイル(stars.html)で動き、外部ライブラリは使わず、できるだけシンプルで、わかりやすい日本語コメントを入れてください。
stars.htmlを上書きできる完全なコードとして出力してください。

共通指示

(4) 画面分割 + Canvasで星を1つ表示

動作内容: ヘッダのボタンはそのままで、メイン画面を左(Canvas)と右(ログ)の2カラムに分割。ボタンを押すたびにCanvasをクリアしてからランダムな位置に星(円形)を1つ描画する。
Done条件：ボタンで黒いCanvasに星が1つ出る(毎回描き直し)。

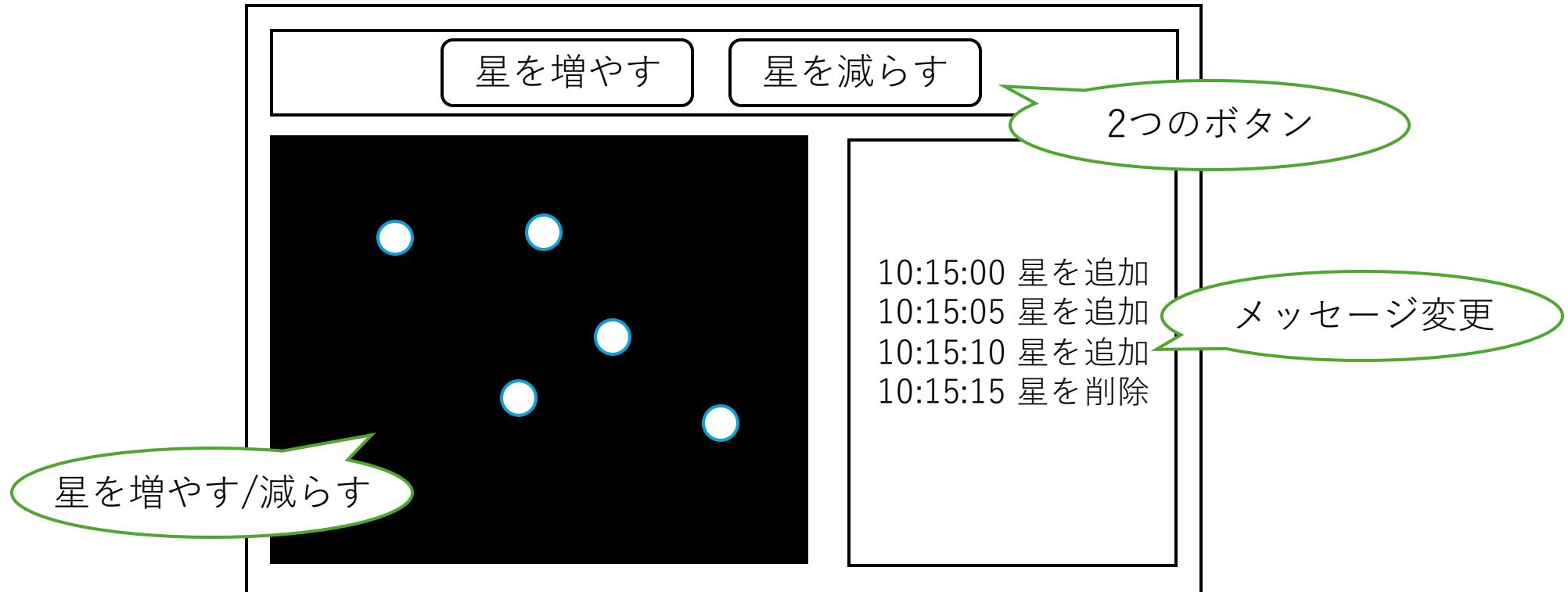
個別指示

追加・修正（例）

- 背景の色を変更する
- 色をランダムにする
- 大きさをランダムにする
- 星の形を五角のスター型にする

Step⑤ ボタンを2つに(星の数を増やす/減らす)

- 「星を増やす」ボタンと「星を減らす」ボタン



Step⑤ ボタンを2つに(星の数を増やす/減らす)

- プロンプトの例

(4)を改造します。

条件として、1ファイル(stars.html)で動き、外部ライブラリは使わず、できるだけシンプルで、わかりやすい日本語コメントを入れてください。
stars.htmlを上書きできる完全なコードとして出力してください。

共通指示

(5) ボタンを2つに(星の数を増やす/減らす)

動作内容: ボタンを2つに増やしてヘッダに「星を増やす」「星を減らす」の2ボタンを配置。

Canvasに表示する星のリストを作り、「星を増やす」ボタンでランダムな位置に星を1つ生成してリストに追加して表示する。一度生成した星の位置は変更しない。「星を減らす」ボタンでリストから星を1つ削除する。

Done条件: ボタンで表示される星の数が増減する。

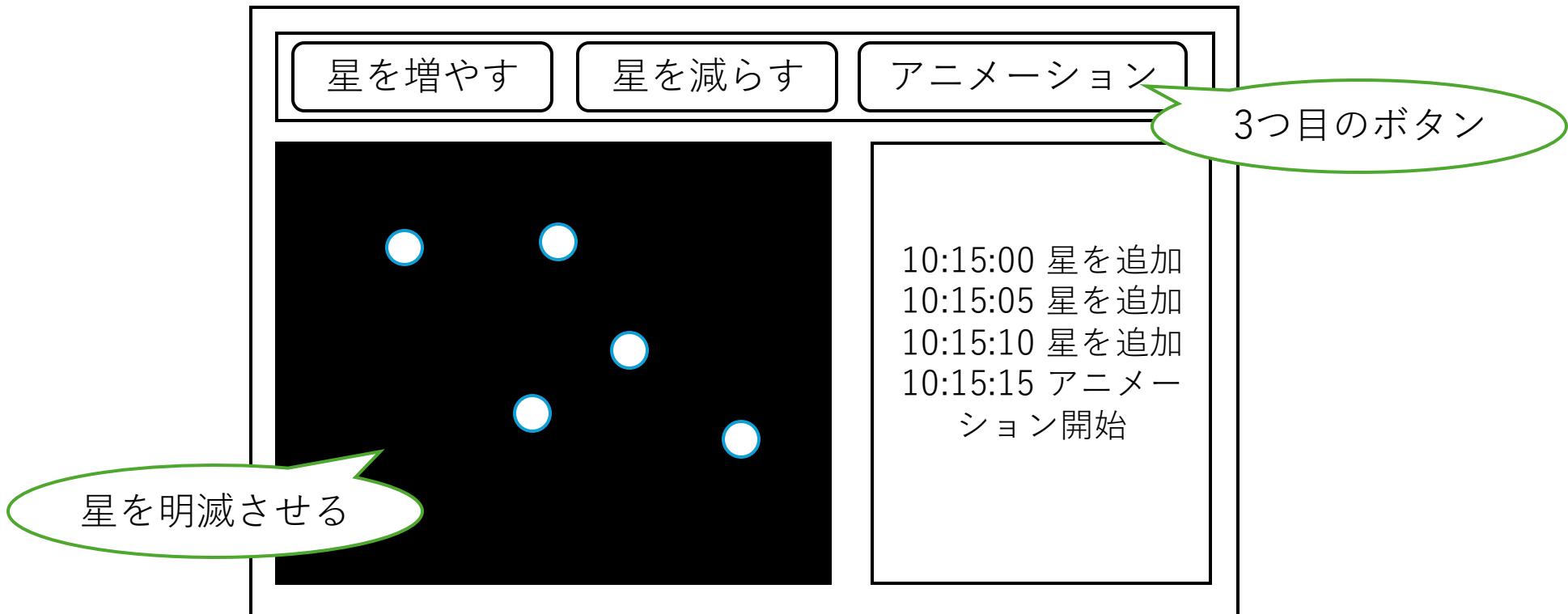
個別指示

追加・修正（例）

- 増減数を10個単位にする
- 星の数を200までに制限する
- クリアボタンをつける

Step⑥ ボタンを3つに(アニメーションのオン/オフ)

- 「星を増やす」ボタンと「星を減らす」ボタン



Step⑥ ボタンを3つに(アニメーションのオン/オフ)

- プロンプトの例

(5)を改造します。

条件として、1ファイル(stars.html)で動き、外部ライブラリは使わず、できるだけシンプルで、わかりやすい日本語コメントを入れてください。
stars.htmlを上書きできる完全なコードとして出力してください。

共通指示

(6) ボタンを3つに(アニメーションのオン/オフ)

動作内容: ボタンを3つに増やして「アニメーション」ボタンも配置。トグル動作で星を明滅させる。

Done条件: アニメーションボタンで星が明滅する。

個別指示

追加・修正（例）

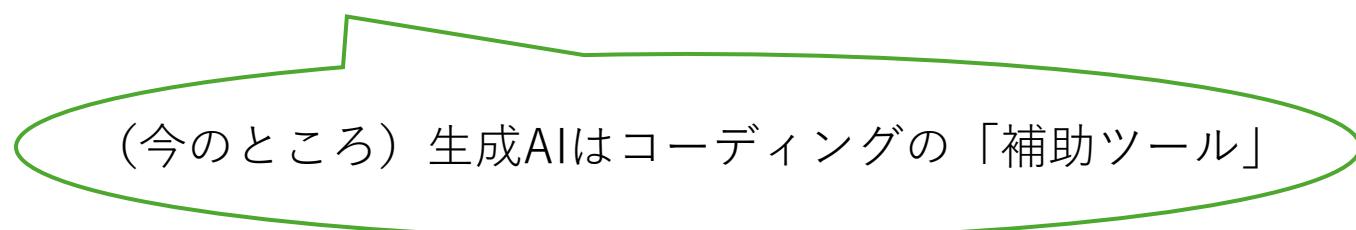
- 明滅のタイミングをランダムに
- 空全体をゆっくり回転させる

まとめ

プログラマは不要になるか？

バイブルコーディングの限界

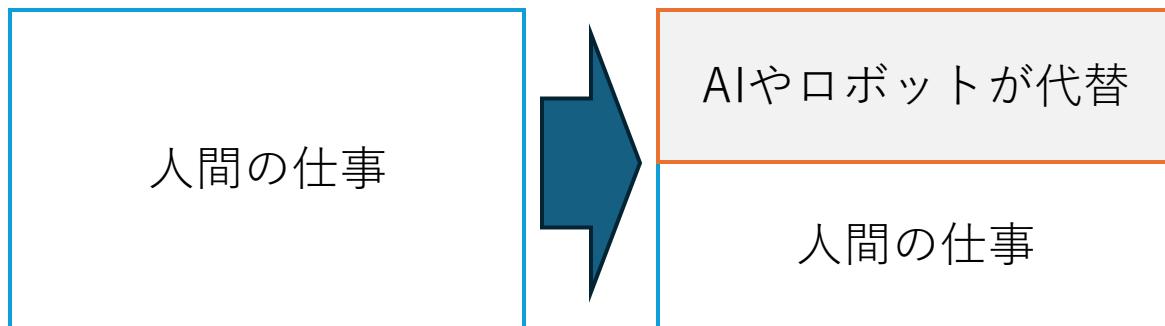
- ・生成されるコードが正しいとは限らない
- ・開発内容の再現性が乏しい、生成するたび違うコード
- ・ブラックボックスと化す、仕様書が書けない
- ・複雑なコードを作ろうとする「ハマる」



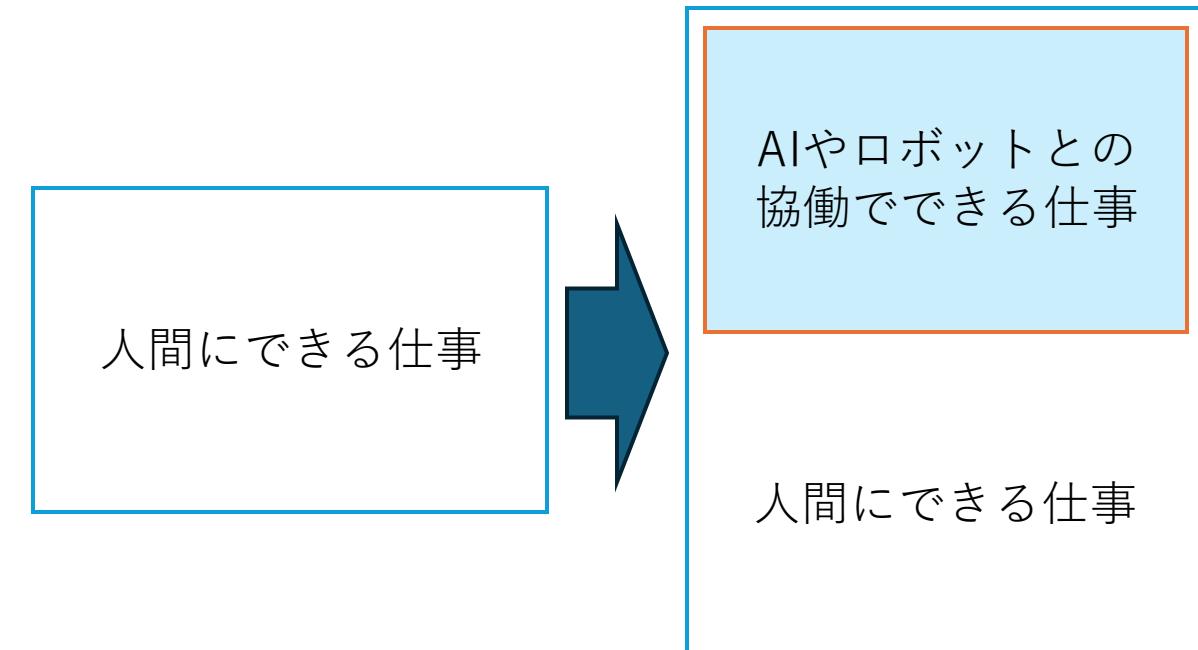
(今のところ) 生成AIはコーディングの「補助ツール」

AI時代に仕事は減る？増える？

- ・仕事が減る（競争）



- ・仕事が増える（共創）



The best way to predict the future is to invent it.

「未来を予測する最善の方法は、自らそれを創りだすことだ。」

Alan Curtis Kay

アラン・カーティス・ケイ

(計算機科学者「パーソナルコンピュータの父」)