餃子を焼くより簡単!?

生成AI×浜松のデータで

アプリをつくる

2Days



浜松市

11/12(木)・11/19(木)

今日のゴール

1.データ連携基盤を知る

2.生成AIに話しかけて、アプリを自分でつくる流れを体験する

★完成することより、流れを体験することが大切です。
うまくいかなくても大丈夫です



本日の進め方

時間	内容
13:30〜	はじめに
13:50〜	ハンズオン環境作成/データ連携基盤・データ活用基礎
15:15〜	仕様書づくり
15:30〜	開発する（ステップ1〜4）
17:00〜	交流タイム
17:20〜	まとめ

ハンズオン環境作成 /
データ連携基盤 - データ活用基礎

今回の環境について補足

開発用の生成AIツールは、いくつかあります

ツール	提供元	使用可能なモデル
OpenCode（今日）	オープンソース	中のAIモデルを各社のものに入れ替えられる
Claude Code	Anthropic社	Anthropicのモデルを使う
Codex	OpenAI社	OpenAIのモデルを使う
Gemini CLI	Google社	Geminiのモデルを使う

＊料金や対応モデルは頻繁に変わります。使う前に、各社の公式情報で確認してください

生成AIツール：チャット用 - 開発用 違い①

	チャット用	開発用（今日はこちら）
たとえ	相談する先生	作業する相棒
AIが見えるもの	貼り付けた内容だけ	プロジェクトフォルダ全体
ファイルへの保存	自分でコピペ/ DLして保存	AIが直接保存
動かす・直す	自分で実行して、結果を報告	AIが実行し、エラーも自分で直す
向いている場面	考える・調べる	作る・直す

今日は、開発用の生成AIにぜひチャレンジしてみましょう！

生成AIツール：チャット用 - 開発用 違い②

	チャット用	開発用（今日はこちら）
始めるのに必要なもの	アカウント登録だけ（ブラウザやアプリですぐ使える）	インストール作業が必要なものが多い
料金	無料プランがあり、気軽に試せる	無料で使える範囲は、限られていることが多い（有料プランや従量課金が前提のものも）

＊料金や対応モデルは頻繁に変わります。使う前に、各社の公式情報で確認してください

今日は、インストール作業も、料金も不要です。運営側で用意しています

ただ、今日の環境は「最低限」です

- 今日は、開発に必要な**環境一式**を**無償で**ご提供
- 一方で、生成AIのモデルは**「軽量版」**

うまくいかなくても、あなたのせいではありません🙆。
エラーも含めて、「生成AIでアプリをつくる流れ」を体験できればOK!

作る前に、AIに相談する

仕様書をつくってみよう

なぜ、仕様書が要るの？

頭の中のイメージは、書き出して初めて、AIに渡せます

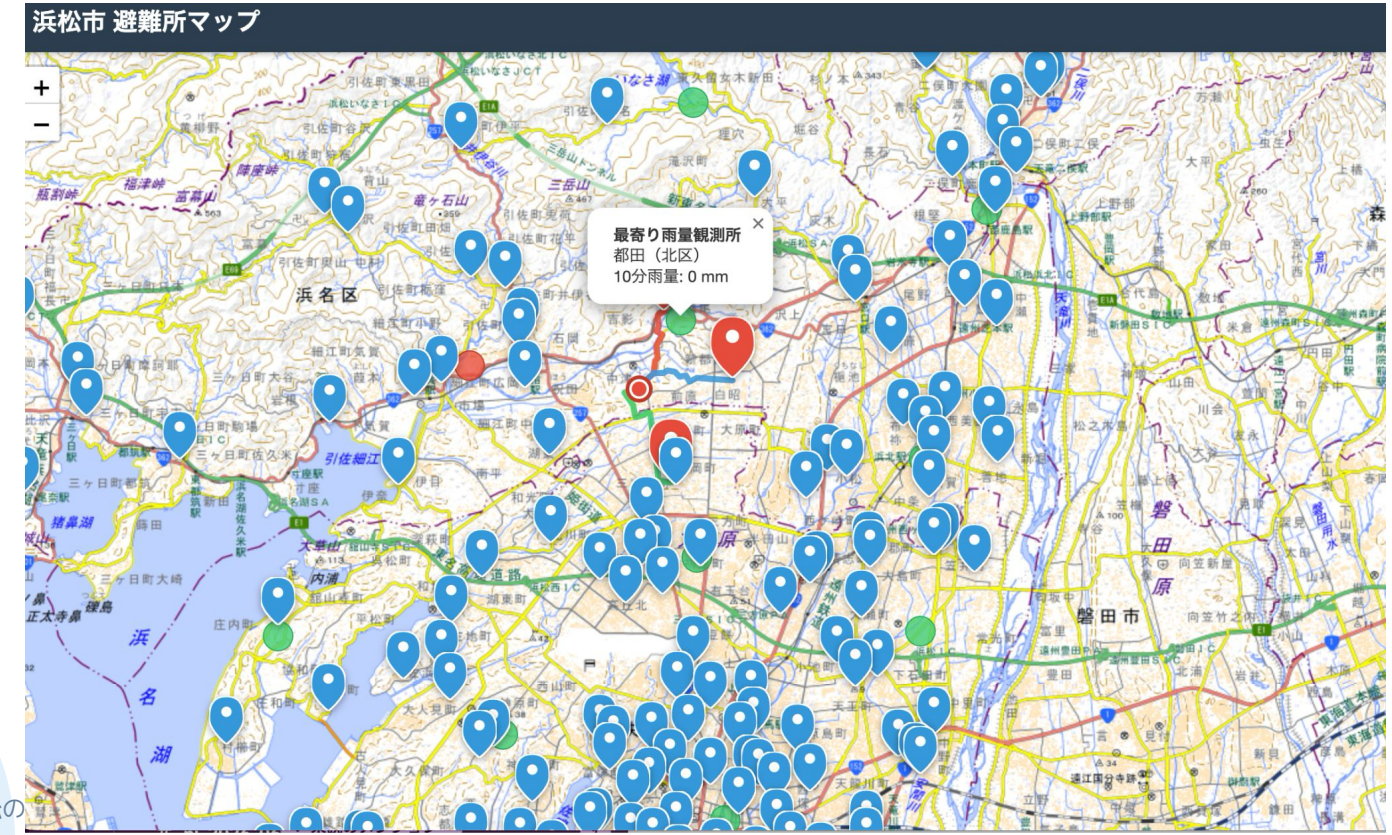


仕様書があると

- 同じ仕様書を渡せば、同じものをつくらせられます（やり直しや、後日の続きにも使えます）
- 質問に答えるうちに、自分が何をつくりたいかが整理されます
- セッション（会話）を新しくした際にも、次のセッションへの「引き継ぎ書」になります

  （セッションが変わると、AIは前の会話を忘れます）

今回の仕上がりイメージ



仕様書に書くこと（例）

大きく３つ

項目	内容
コンセプト	何のアプリか（1文）
使うデータ	どんなデータを使うか
画面	画面に何が出るか

記入例

```
## コンセプト
避難の際、雨量と最寄りの避難所がすぐ分かるアプリ

## 使うデータ
避難所データ・雨量データ

## 画面
- 地図に避難所をマーカー表示
- 雨量を色（青→赤）で重ねる
- 地図をクリックして現在地を決め、最寄り3か所へのルートを表示
```

PlanとBuild

モード	役割	今日の使いどころ
Plan	考える・相談する（ファイルは変えない）	仕様書の壁打ち
Build	作る・保存する	仕様書の保存、アプリづくり

- 切り替えは **Tab キー**
- 今、どちらのモードかは、入力欄の上の表示で確認できます



★Planモードでは、ファイルを保存できません。仕様書を保存する時は、Buildに切り替えます

仕様書をつくる①：AIに質問してもらう

モード：Plan

次のプロンプトを、貼って送信します

避難の際、雨量と最寄りの避難所がすぐ分かるアプリを作りたい。初学者向きにシンプルな機能前提です。
仕様書をつくるために必要な事項を一つずつ質問して。質問数は１０問以下でお願い。

- AIが、1つずつ質問してきます。短く答えてください
- 答え方で、できあがる仕様書が変わります

仕様書をつくる①：AIに質問してもらう

- AIに今回使うデータを聞かれたら、下記を指定してください（配布済み）
  - 避難所：data/hamamatsu/shelters_hamamatsu.csv
  - 雨量：data/hamamatsu/rain_hamamatsu.csv
    - ＊10分雨量と60分雨量の項目があります。今回は10分雨量項目を使用してください。

仕様書をつくる②：書き出してもらう

モード：Build（Tab で切り替え）

質問が終わったら、次のプロンプトを送ります

> ここまでの内容を、spec.md というファイル名で保存してください

- 左のファイル一覧に spec.md ができれば完了です
- ファイル名はなんでもOKです！
- .md というのは、MarkDownファイルのことです。AIが読みやすいテキスト形式の書き方です。

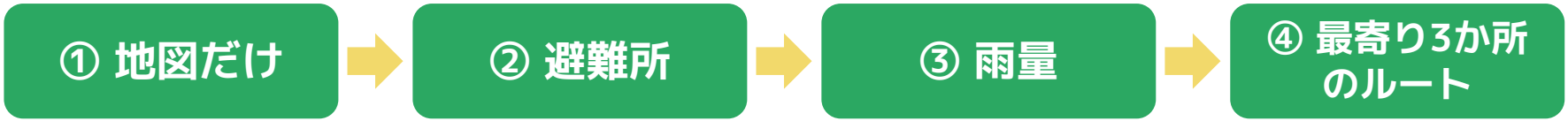
仕様書ができたら、新しいセッションへ

- **/new** と入力して、Enter
- なぜ？
  - 同じセッションで会話を長く続けると、AIの記憶の上限を超えてしまう場合があります。
  - 仕様書があれば、新しいセッションでも、要旨を引き継げます
- 作業がひと段落したら、セッションを切り替えるのがよいです
- 過去のセッションを選ぶ際は　**/sessions**

開発にかかります🙌！

大きく４ステップです

4ステップでつくります



毎回の型

1. **/new** で、新しいセッション
2. プロンプトを貼って送信
3. 動作確認

→**ローカルサーバーで（後述）**

コツ

- 一度に全部を頼まず、動いたことを確かめながら、1つずつ「何がしたいか」を足していきます

ステップ１：地図を表示する

まず小さくやってみよう

ステップ1：地図だけ表示する　の完成イメージ

- 完成イメージ：浜松の地図だけが出ます
- 最初に **/new** で、新しいセッションにします

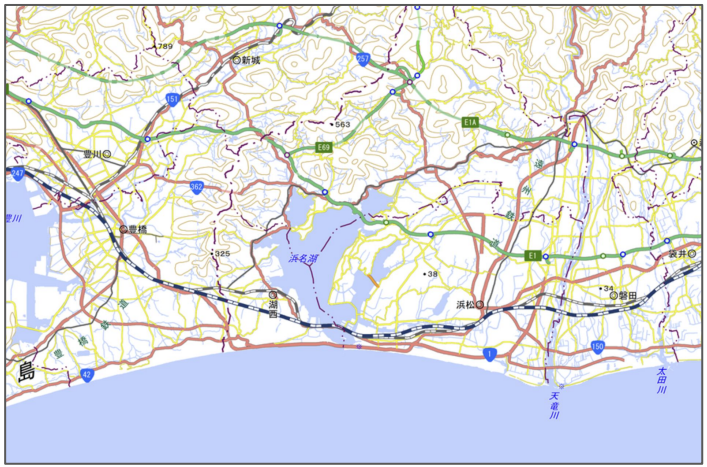
ステップ1-a：地図だけ表示する

モード：Build

spec.mdの内容に沿って、まず地図だけを表示させて。他の機能はまだ作らないで

ステップ1-b：ローカルサーバーで確認

- 次頁の手順でローカルサーバーを起動する
- 地図が出れば、OKです



ローカルサーバーを起動しよう（1回起動しておけばOK）

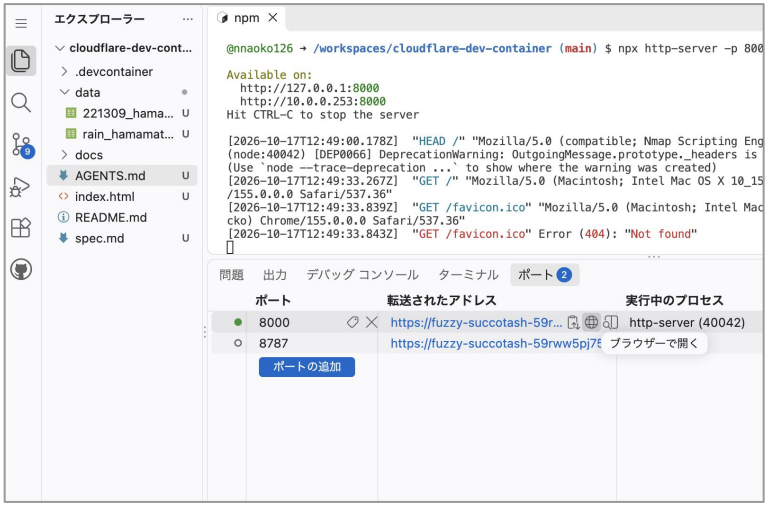
1. 新しいターミナルを開く

2. 次のコマンドを実行する

```
npx http-server -p 8000
```

3. ブラウザで確認する

・Codespaces：「ポート」タブの8000の、
地球儀アイコンをクリック



★実行した後は、このターミナルを閉じないでください（閉じると、サーバーが停止します）

サーバーって何？　ローカルサーバーって何？

サーバー

- ブラウザから「このページ・このデータをください」と頼まれて、渡す係です
- たとえ：お客さん（ブラウザ）の注文を受けて、料理（ページやデータ）を出すお店の人
- 普段見るWebサイトも、遠くのサーバーから届いています

ローカルサーバー

- その係を、自分の作業環境の中に呼んで、自分だけに渡してもらう、お試し用の小さなサーバーです

サーバーでつまずいたら

こんな時	こうする
AIが「サーバーが起動していないようです」と言った	前のページをやり直す
起動したはずのターミナルが見当たらない	ターミナルを開き直して、同じコマンドを実行
address already in use と出た	直そうとせず、番号を変える（8001、8002…）。確認するURLの番号も変わる

ステップ2：避難所を表示する

ステップ2：避難所を表示する　の完成イメージ

- 完成イメージ：避難所が、青いピンで出ます
- 最初に **/new** で、新しいセッションにします
- 使うデータ：避難所のCSV（配布済み）
    - data/hamamatsu/shelters_hamamatsu.csv

ステップ2-a：避難所を地図に表示する

モード：Build

spec.mdの内容に沿って、避難所データを地図に表示させて。
他の機能はまだ作らないで。

ステップ2-b：確認

- ページを強制再読み込み（**Ctrl+Shift+R**、Macは**Cmd+Shift+R**）

- 青いピンをクリックして動きを確認してみましょう



余力のある方へ：吹き出しを整える

先に進めた方は、挑戦してみましょう（必須ではありません）

マーカーをクリックした時の吹き出しに、施設名・住所・電話番号（データにあれば）を見やすく整えて表示してください。空の項目は表示しないでください。地図の下地（タイル）は変えないでください。

うまくいかない時は、まず画面を見てAIに聞く

画面の様子	こうする
画面の上に、赤い帯が出た	その文字をコピーして、AIに貼る
赤い帯も出ず、真っ白	F12 → 「Console」タブの赤い文字をコピーして、AIに貼る
直らない	/new で新しいセッションにして、やり直す

貼れる一言：次のエラーが出ました。原因と直し方を教えてください。
（エラー文をそのまま貼る）

ステップ3：雨量を表示する

ステップ3：雨量を表示する　の完成イメージ

- 完成イメージ：雨量サークルが表示されます。
- 最初に **/new** で、新しいセッションにします
- 雨量データは　data/hamamatsu/rain_hamamatsu.csv を使用。
  - 観測地点ごとの、10分雨量・60分雨量のCSVです。
  - 今回は10分雨量の項目を使います。

ステップ3-a：雨量を地図に重ねる

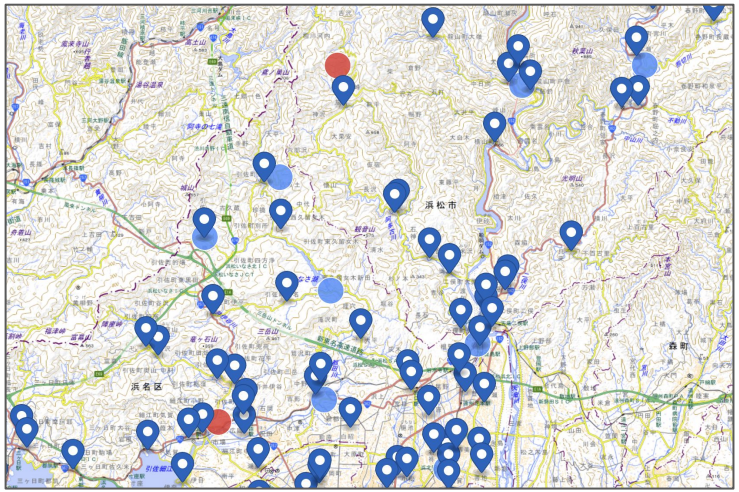
モード：Build

spec.mdの内容に沿って、雨量データを地図に重ねて表示させる機能だけ追加して。
他の機能はまだ追加しないで。

ステップ3-b：確認

- ページを強制再読み込み（**Ctrl+Shift+R**、Macは**Cmd+Shift+R**）

- サークルの大きさがいまいちだったら「**サークルを○倍の大きさにして**」と指示してみましょう。
- 雨量に応じて青→黄色→赤の円が表示されるよう、「**サークルの色は、50mm以上だったら赤、30mm以上だったら黄色にして**」等、指示してみましょう。



ステップ4：最寄り避難所ルート表示

ステップ4：最寄り3避難所へのルートを出す　完成イメージ

- 完成イメージ
  - 地図をクリック → 緑の現在地ピン
  - 最寄りの3避難所への、ルート線
- 最初に **/new** で、新しいセッションにします

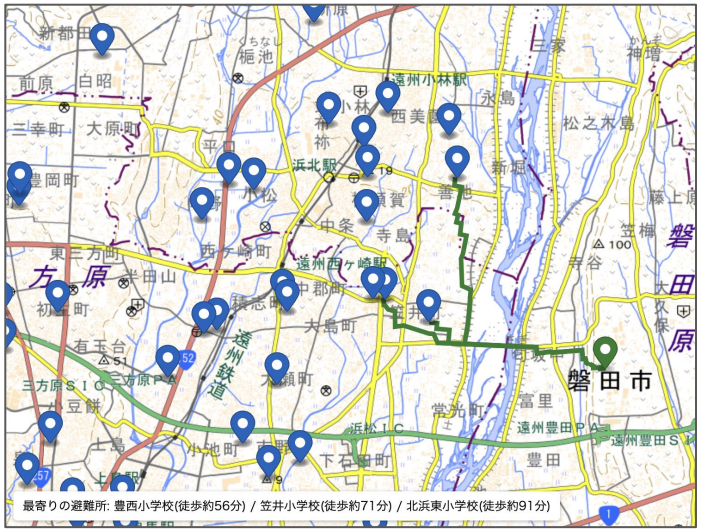
ステップ4-a：クリックした場所から、最寄りの3避難所へ

モード：Build

spec.mdの内容に沿って、地図をクリックしたところを現在地にして、そこから最寄りの3避難所の徒歩ルートを表示させて

ステップ4-b：確認

- ページを強制再読み込み（**Ctrl+Shift+R**、Macは **Cmd+Shift+R**）
- 地図をクリックして動きを確認してみましょう



完成です！🙌 お疲れ様でした！

よくある質問

よくある質問

質問	こうする
ターミナルのコマンドプロンプトが戻ってこない	プロセス実行中のため（ローカルサーバーなど） 強制終了させる場合は、**Control+C**。ターミナルは **exit** で終了
opencodeの出力内容をコピーできない	仮想マシン環境のためコピーできません。
opencodeでスクショが貼れない	仮想マシン環境のためコマンド欄には貼れません。ファイルとしてアップロードすることはできます。
opencodeでファイルをアップロードするには？	Codespacesのエクスプローラーパネルにドラッグ＆ドロップしてください

よくある質問

質問	こうする
AIの応答が急に英語になった	チャット欄に「日本語で」と入力すれば戻ります
頼んだ反応と違う・保存されない	入力欄上のモード名を確認。Planのままでは? **Tab** で切替
AIに何度直させても同じ理由で失敗し続ける	対象ファイルの現在の内容を全文コピーして、AIへの依頼メッセージに貼り付けてから再度頼んでください （別のAIに聞くのも一手）

よくある質問

質問	こうする
OpenCodeで Unauthorized エラーが出た	ターミナルで　**opencode auth login https://opencode-for-events.tokyo-odh-002.workers.dev** を実行後、opencodeを起動してください
opencodeを終了させたい	**/exit**
ローカルサーバーの落とし方	該当ターミナルを**Ctrl+C**で閉じる。 もしくは、ターミナルから　**pkill -f http-server**　を打つ

おまけ

文字コードについて

文字コードの話

- 自分でデータを探して使う時は、そのデータの文字コード、列の並び、件数を把握しておきましょう。
  - エディターパネルの右下に文字コード（UTF-8やShift-JIS）が表示されます
- 文字コードによって、文字化けやエラーが出ることがあります
- エラーが発生した場合は
  - エラーを、AIに伝える
  - 直し方は、AIが教えてくれます

使うデータで、最初に確かめること

1. 文字コード
2. 列の並び
3. 件数

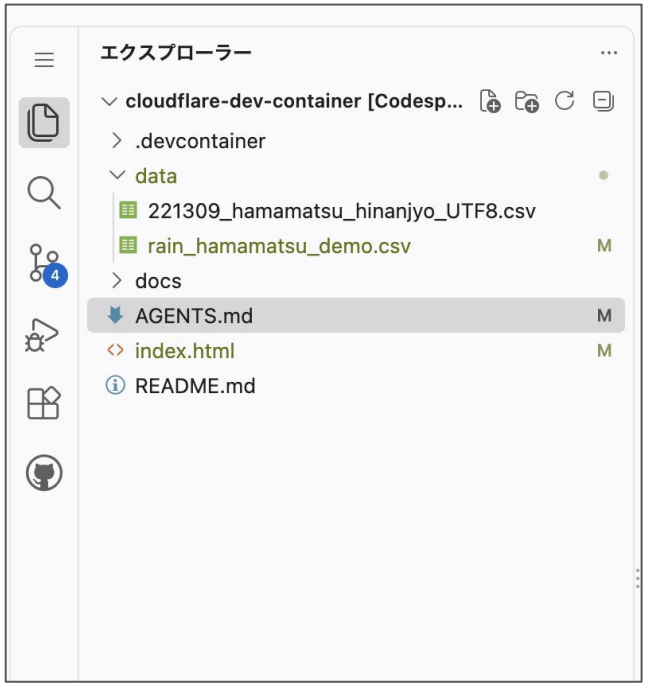


参考1

AGENTS.md について

「AGENTS.md」って、何？

- AIへの「常設の指示書」です
  - 作業フォルダに置いておくと、AIが毎回、最初に読んでくれます
- たとえ：新しく来た作業員に渡す「ルールブック」。毎回、口で説明しなくて済みます
- 今日は、配布済みです。自分で作る必要はありません



AIがどんなルールを認識しているか、
以下のプロンプトで確認できます。

このフォルダのルールを教えて。

「AGENTS.md」はどう作る？

- 必須ではありません。なくても動きます。
- もし、あなたがこれからつくる時も、自分で一から書かなくて大丈夫。
  以下のプロンプト等で、AIに作ってもらえます。

このフォルダは○○○○用です。
このフォルダに必要なAGENTS.md を提案して。

他の生成AIでも、同じファイル名？

ツール	指示書のファイル名
OpenCode（今日）	AGENTS.md
Codex	AGENTS.md
Claude Code	CLAUDE.md
Gemini CLI	GEMINI.md

- 名前は違っても、役割は同じ（AIへの常設の指示書）
- チャット型にも、近い機能があります（カスタム指示、プロジェクトの指示など）
- 2026年10月時点の情報です。ツールの更新で変わるので、使う前に公式ドキュメントで確認してください

参考2

つくったアプリを公開するには？

つくったアプリを、公開するには？

方法	内容	位置づけ
① ポートの公開	Codespacesの「ポート」を公開にして、URLを共有	**お手軽**
② wrangler dev ＋ tunnel	手元の環境から、一時的に公開	発展
③ wrangler deploy	Cloudflareに、本番として公開	発展

公開のしかたも、AIに聞けます

つくったアプリを、他の人にも見せたいです。方法を教えてください

「見せたい」と伝えて、方法を聞く体験で十分です

参考3

コードの変更履歴を保存するには？

コードの変更履歴を保存する仕組み（Git）

- Git ＝「履歴地点」に戻れるセーブポイントの仕組み
- Codespaceでは、最初から使えます（git initは実施済み）

履歴を保存する

- 履歴を保存したい際に、ターミナルで以下２つを打つ

```
git add .
git commit -m "ver1_初版"  ←””は内容に合わせて変える
```

- AIにお願いしてもできます

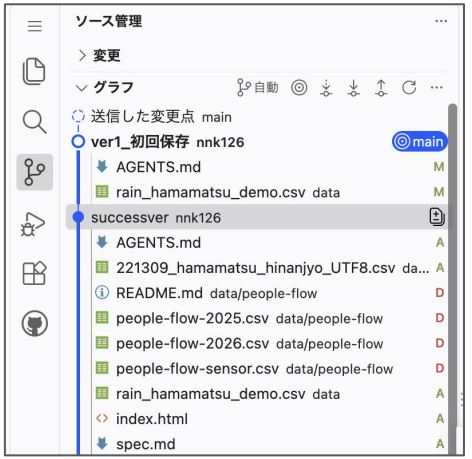
```
gitで保存して
```

※以下のメッセージがでますがcommit自体はできます。今回の環境ではpushできません。
pushuしたい場合はリポジトリをご自身の環境にforkしてCodespaceを作成してください。
You don't have write access to the kzkski/cloudflare-dev-container repository, so you cannot push changes to it.
To obtain write access we will point this codespace at your fork of kzkski/cloudflare-dev-container, creating that fork if it doesn't exist.

コードの変更履歴を保存する仕組み（Git）

履歴を見る

Codespaceの「ソース管理」ビューで、保存した履歴が見られます



Gitの細かな仕組みは、「Gitの使い方について教えて」とAIに聞いてみてください