

Vibe Coding 入門

～ AIと共に創る、次世代のコーディングスタイル ～



アジェンダ (60 分)

1. Vibe Coding とは？ (10 分)

- 新しいコーディングの定義
- 人間と AI の役割分担

2. なぜ流行っているのか (5 分)

- AI の進化と生産性の変革

3. 具体的なやり方 (15 分)

- AI への「指示」の技術
- 音声 vs テキスト

4. ツールの紹介 (10 分)

- コーディング AI の「三種の神器」

5. ワークショップ (15 分)

- お題：「AI にランディングページを作らせよう」

6. Tips & まとめ (5 分)

1. Vibe Coding とは？

AI 主体・人間ディレクションのコーディング

- 人間が1行1行タイピングするスタイル... ではない。
- 人間が「どうしたいか (Vibe)」を音声やテキストで指示し、
- AI が主体となってコードを生成・修正する開発スタイル。

1. Vibe Coding - 役割の 変化

従来

- 人間: タイピスト 兼 設計者 兼 デバッガー
- ツール: エディタ、コンパイラ（受動的）

Vibe Coding

- 人間: ディレクター (監督)、アーキテクト、レビュアー
- AI: メインプログラマー、タイピスト (能動的)

人間の役割は「How (どう書くか)」から「What (何を作るか)」と「Why (なぜ作るか)」へシフトする。

2. なぜ流行っているのか

1. AI の飛躍的進化

- LLM (大規模言語モデル) が、複雑なコードの意図を理解し、高品質なコードを生成できるようになった。

2. 生産性の爆発的向上

- タイピングのボトルネックから解放される。
- ボイラープレート（お決まりのコード）は AI が一瞬で生成。
- 人間は「設計」と「仕様」の伝達に集中できる。

3. 「思考の速度」での開発

- 「こうしたいな」と思った "Vibe" を、即座に AI に伝えて形にできる。
- アイデア → 実装 のサイクルが極端に短くなる。

3. 具体的なやり方： AI への「指示」

AI は「察する」のが苦手。

良い Vibe（指示）は、**具体的かつ文脈（コンテキスト）** を含んでいる。

✗ 悪い指示 (Vibe が伝わらない)

- 「ユーザー一覧を表示して」
- 「ここ、なんかバグってる」
- (音声で) 「ボタン作って」

○ 良い指示 (Vibe が伝わる)

- 「`users` 配列を `map` で展開し、Tailwind CSS のカード型コンポーネントで一覧表示して。各カードには名前とメールを表示して」
- 「この `useEffect` が無限ループしている原因を特定し、修正案を出して」
- （音声で）「`signIn` 関数を実行する、青色のプライマリボタンを実装して」

3. 具体的なやり方：反復的な指示

「AI との対話」 がコーディングになる。

1. 指示 (人間): 「ログインフォームを作って」
2. 生成 (AI): (コードを生成)
3. レビュー (人間): 「OK。でもパスワード忘れのリンクも追加して」

4. 修正 (AI): (リンクを追加)
5. レビュー (人間): 「デザインがイマイチ。Tailwind CSS でモダンにして」
6. 修正 (AI): (CSS を適用)
7. ...

AI を「超優秀なジュニア開発者」として扱い、レビューと指示を繰り返して育てる。

4. ツールの紹介

1. GitHub Copilot

- コーディングの相棒
- エディタ (VS Code) に統合され、コーディング中にリアルタイムでサジェスト。
- **Copilot Chat**: テキストで「このコードのバグは？」と質問できる。

2. Cursor

- AI ネイティブエディタ
- AI がプロジェクト全体を読み込み（ でファイル指定）、広範囲なリファクタリングや機能追加が可能。
- まさに「AI に指示して作らせる」体験に特化。

3. Claude Code / Gemini CLI / Codex

- コマンドライン AI エージェント
- コマンドライン(Terminal など)にインストールして、PC 上で直接動かす AI
- 与えられたタスクに必要なことを、自分で分析して、順序立て、実効する。

4. ChatGPT / Gemini / Claude

- 壁打ち & 設計の「相談役」
- 「この機能、どんな技術スタックがいい？」
- 「Go と Rust、どっちが向いてる？」
- 「このエラー、どういう意味？」
- 特定のアルゴリズムや複雑なスニペットを丸ごと作らせるのに強い。

5. ワークシヨップ (15分)

お題：「AI に自己紹介 LP を作らせよう」

- ゴール:
 - AI への指示（テキスト）だけで、簡単な自己紹介ランディングページ (LP) を作成する。

- ツール:

- 先ほど紹介したものでも、それ以外でも
- Cursor は学生は 1 年無料
- CodePen (実行環境)
- もしくは VSCode の LiveServer を使うとプレビューを見れる

5. ワークショッップ - Vibe を伝えよう！



Let's Vibe Coding!

AI への指示 (プロンプト) 例:

“

あなたはプロの Web デザイナーです。
HTML, CSS, JavaScript を全て 1 つの HTML ファイルに
まとめて、以下の要件で自己紹介ページを作ってください。

1. **テーマ**: ダークモードで、モダンかつミニマルなデザイン。
2. **ヘッダー**: 私の名前「[あなたの名前]」と「Software Developer」という肩書き。
3. **セクション**: 「私について」「スキル (React, Go, AWS をリスト表示)」「趣味」の 3 つ。
4. **インタラクション**: スキルリストにマウスをホバーすると、少し拡大するアニメーションを CSS で追加して。

”

Tips:

- まず生成させ、実行してみる。
- 気に入らない箇所を「もっとこうして」と追加指示する。

6. Tips & まとめ



AI 使いこなし術 (Tips)

1. AI を「副操縦士」として信頼し、人間は「機長」として最終責任を持つ。

- AI は平気で嘘（ハルシネーション）をつき、バグも生みます。
- 生成されたコードは必ずレビューする！

2. 「なぜ」を説明させる。

- 「なぜそのコードを選んだの？」と聞くと、AI の思考が分かり、学習にもなる。

上級者編(MCP サーバー)

MCP サーバーとは AI が使えるように設計された API のようなもの

- 様々な種類があり、これらを上手く活用することで、AI の苦手な部分を上手くカバーしたり、さらなる効率化を図ることができる



まとめ

- **Vibe Coding** は、AI に指示し、AI が実装するスタイル。
- 人間の役割は「タイピング」から「ディレクション」と「レビュー」へ。
- 「いかに上手く AI に Vibe（意図）を伝えるか」 が、これからの開発者のコアスキルになる。

Q&A

ご清聴ありがとうございました！