Add Eight

AI 駆動型開発ガイド

AI と開発者が、安全に・実践的に開発を進めるための標準ガイド



Version 0.3

1.2 AI と開発者の役割



AI は作業を進める役、開発者は重要な判断と最終確認を行う役、と考えると分かりやすくなります。

1.3 AI が「完了」と言っても完成ではない

AI は間違えることがあります。

たとえば、

- 仕様を勘違いしたまま実装する
- 依頼していない部分まで変更する
- テスト自体が間違っている
- 本来直すべき不具合とは別の問題を直す
- 「問題ありません」と報告したコードに問題が残る

ことがあります。

そのため、**AI の完了報告ではなく、開発者の確認をもって完成とします。**

Add Eight ルール

- AI に作業を任せても、最終判断まで任せない。
- AI の回答を 100%正しい前提で扱わない。
- AI が「完了」と言っても、開発者が確認するまで完成ではない。

第 4 章　良いプロンプトを作る

まずは対話で理解しよう



新人　ハルト

良いプロンプトって、とにかく長く書けばいいんですか？

先輩　ミナト

そう思う人は多いよね。でも、大事なのは長さより中身なんだ。目的、仕様、変更してよい範囲、完成の条件、それから AI が自分で判断してよい範囲をはっきりさせることが重要だよ。



この章で学ぶこと

AI へ何を伝えると、意図した作業を進めやすくなるかを学びます。

4.1　プロンプトとは

AI への指示文です。

良いプロンプトは「長い文章」ではありません。**AI と開発者の判断範囲が明確な指示**です。

4.2　基本の 8 項目

必要に応じて次を含めます。

1. 目的
2. 現在の状態・調査結果
3. 対象範囲
4. 実装内容・確定仕様
5. 期待する動作
6. 制約・変更禁止範囲
7. テスト・レビュー
8. 完成条件

たとえば「タスクに期限を追加する」場合、

- **目的**：タスクに期限を設定できるようにする
- **対象範囲**：タスクの登録・更新・表示

付録 F　社内の実体験・失敗事例

実際に社内で AI を使って開発したときの失敗と、その後に決めたやり方をまとめています。同じ失敗をしないように、作業の前に読んでください。

F.1　見た目の一部だけ直すつもりが、関係ない場所まで変わった

やろうとしたこと	画面の特定の場所の見た目を、AI に変更してもらった。直したい場所は、画面の画像を見せて伝えた。
起きたこと	指定した場所だけでなく、関係のない場所まで変更された。全体のバランスも、AI が自分の判断で決めて作り直していた。
原因	画像をそのまま渡したため、AI は画像に写っている範囲全体を修正してよいと受け取った。変更してよい範囲と、画像の中のどこを直すのかを、細かく伝えていなかった。
今のやり方	画像に赤線や枠を付けて、直す場所を示す。プロンプトにも「赤枠の部分」「赤線の部分」と書き、画像と文章の両方で直す場所が伝わるようにする。
関係する章	第 4 章　良いプロンプトを作る／第 7 章　AI に実装を依頼する

悪い例（当時の依頼）

```
この画像の部分の見た目を直してください。
（画面の画像を添付）
```

良い例（今の依頼）

```
添付した画像の、赤枠で囲んだ部分だけを修正してください。
修正内容：赤線を引いたボタンの文字を大きくし、背景色を濃い青にする
赤枠の外の部分は変更しないでください。
余白や配置など、画面全体のバランスも今のまま変えないでください。
修正する前に、変更するファイルと変更内容を報告してください。
```

ポイント

- 画像で伝えるときは、「どこを」「どう直すか」「どこは変えないか」を、画像の印と文章の両方で伝える