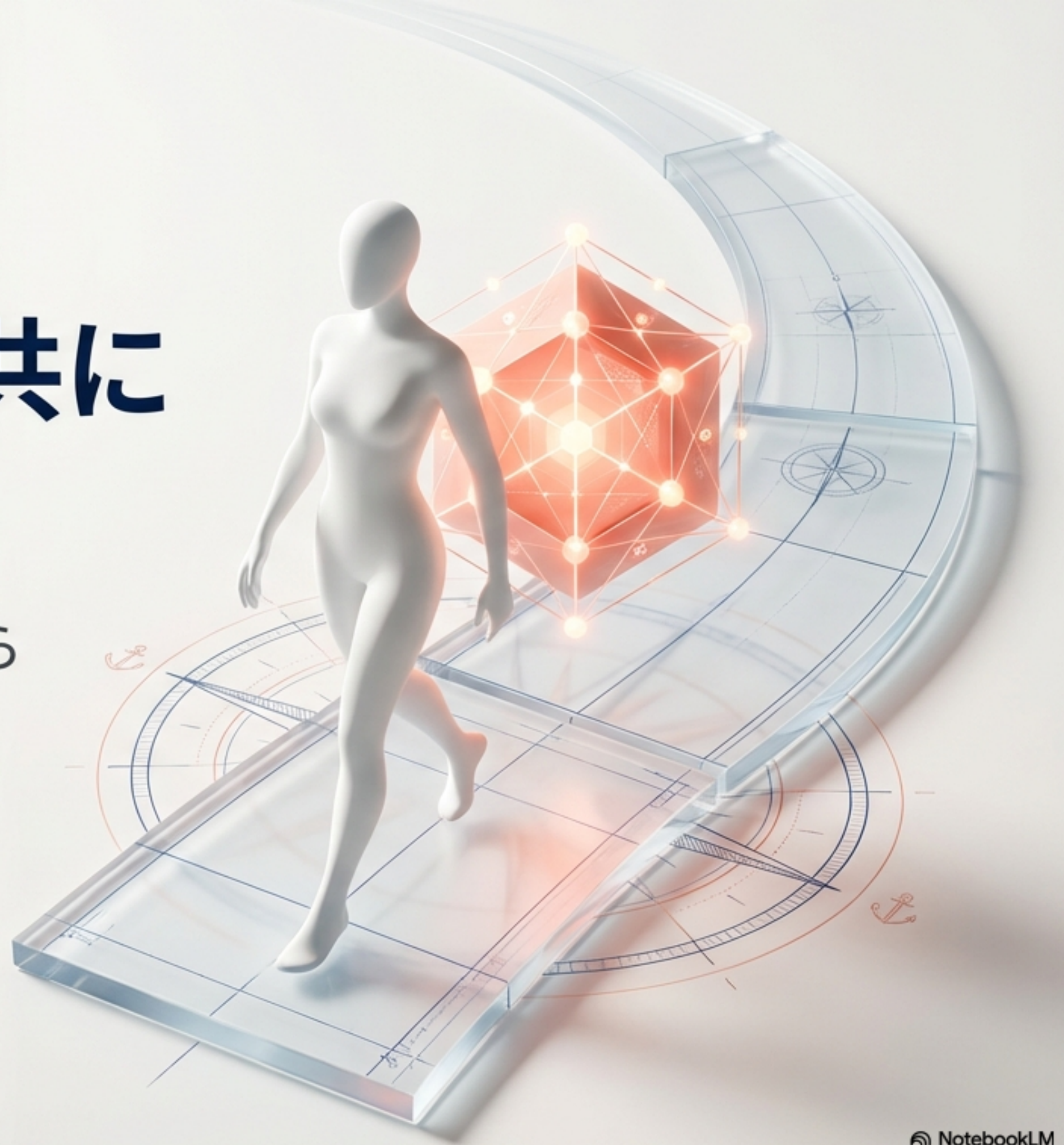


AIエージェントと共に 創る新時代の開発

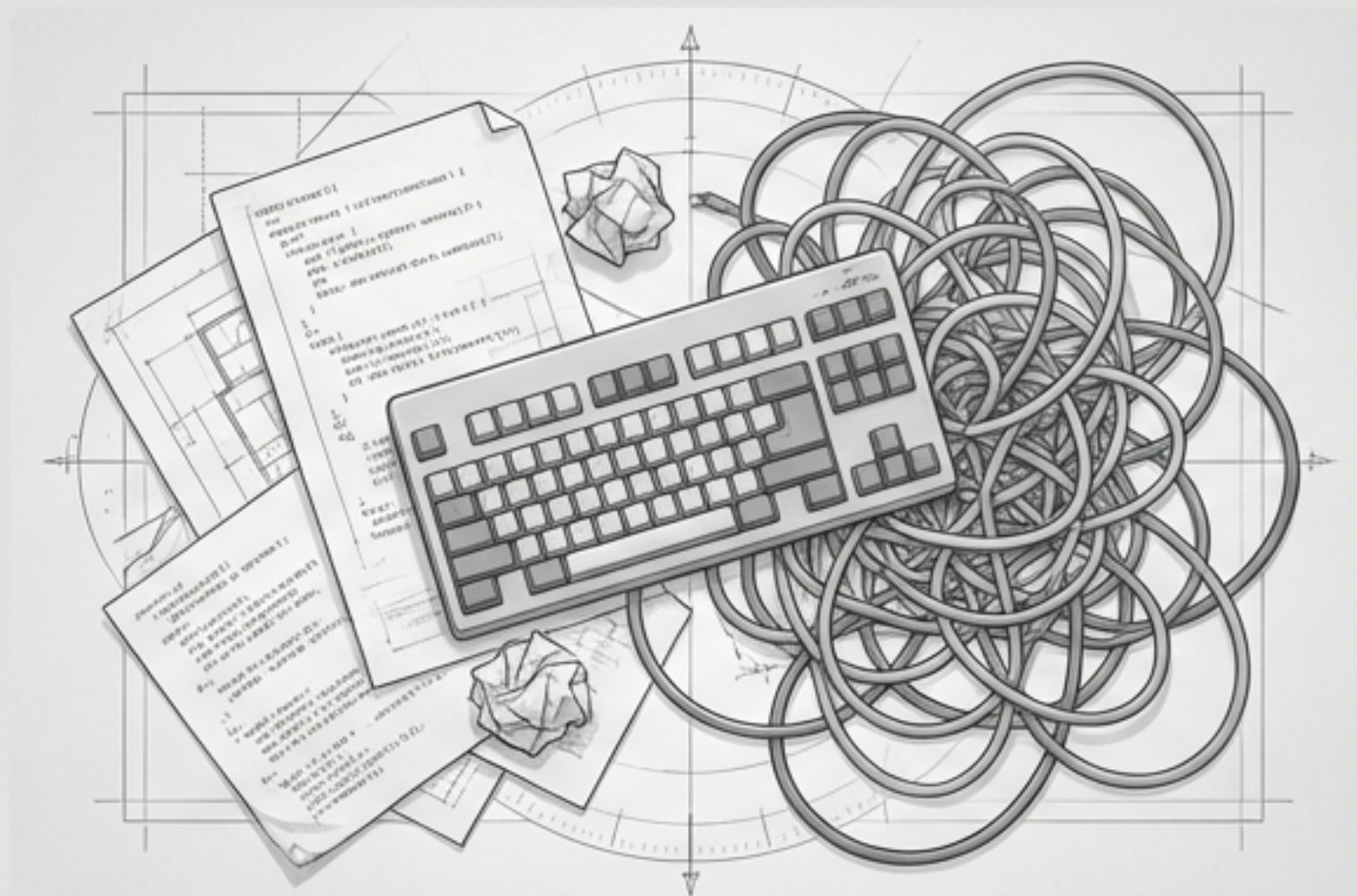
- Cursorを活用した「コーダー」から「キャプテン」への進化
- AI支援開発に特化した短期集中プログラム（全4回）



開発のアプローチが根本から変わる

左側（従来）

全てをゼロから手作業で記述する。シンタックスエラーや些細なバグに時間を奪われる孤独な作業。



右側（AIエージェント前提）

Cursorを相棒としたペアプログラミング。意図を伝え、AIがコードを生成し、人間がそれを検証・洗練させる。



Key Takeaway: あなたの仕事は「コードを書くこと」だけではありません。

あなたの新しい役割は「指揮官（キャプテン）」

AIの立ち位置

非常に優秀だが、文脈を持たない
「ジュニアアシスタント」。

あなたのミッション

1. 何をAIに任せるか（委譲）を決定する。
2. 何を自分で判断・検証するか（レビュー）を見極める。

生成されたコードや提案の「意図を理解」し、
「修正」する力が、今後の最重要スキルになります。



本カリキュラムが目指す3つの到達点



Pillar 1: 意図の理解と検証

AIの出力を盲信せず、生成されたコードの正しさや背景にある意図を読み解き、確信を持って検証できる。



Pillar 2: 適切な指示と修正

タスクを分解し、正確な文脈を渡し、AIの誤りや不適切な部分を的確に修正できる。



Pillar 3: 機密・倫理の遵守

社内ルールを守り、機密情報を保護しつつ、AIの限界（ハルシネーション等）を前提とした安全な運用ができる。

キャプテンの座に就くための準備

必要な前提知識

プログラミングの基礎（変数・制御・関数）の理解。テキストエディタ・コマンドラインの初歩的な操作スキル。

開発環境

AI支援開発ツール（Cursor）へのアクセス権限。

ルールの確認

社内のAI利用ポリシー、およびハルシネーション防止ルール（Team Rules）への同意と理解。




Message: 既存のプログラマーカリキュラムの基礎があれば、準備は完了しています。

AI開発を制する6つのコアスキル



盲信の代償：AIの「もっともらしい嘘」を見抜く

```
1 import fake_api as fa
2
3 # AI-generated success
4 fa.fetch_data(user_id, password)
5
6 # AI generated illustration
7 fa.fetch_data(user_id, password)
8
9 func conton(){
10     if (data == alembic) {
11         println(user_data)
12     }
13     return fa.fetch_key_data()
```



fa.fetch_data(SECRET_KEY, user_data)
WARNING: SECURITY VULNERABILITY
- FAKE API CALL



ハルシネーション（幻覚）への警戒

AIは存在しないAPIや、古いライブラリの仕様を自信満々に提案することがあります。常に「**実在確認の習慣**」を持つことが求められます。



セキュリティの絶対防衛線

機密情報（認証キー、個人情報、非公開アルゴリズム）は絶対にプロンプトに含めない。

Message: 指揮官の最大の責務は、コードを書くことではなく、品質と安全を「保証」することです。

1~2週間で完結する、4ステップの航海図

Session 1

導入と基礎

(プロンプト体験とセキュリティ)

Session 3

検証と防衛

(ハルシネーション対策とテスト)

Session 2

設計と文脈

(明確な指示とコンテキスト連携)

Session 4

実務接続

(リファクタリングとコードレビュー)

短期間で集中的に学び、すぐに実務でAIを活用できる状態を目指します。

第1回：AIツールの導入とプロンプトの基礎

Focus

Cursorの環境導入と、プロンプトによるコード生成・補完の初回体験。

Core Action

生成されたコードを単にコピーするのではなく、「読解」し「実行」して動きを理解する。

Critical Rule

プロンプトに機密情報を絶対に入れないことの徹底。

Deliverable

導入手順メモ、簡単なプロンプト例と生成コードの対応メモ。



第2回：プロンプトの設計と「文脈」の渡し方

Focus

巨大なタスクを分解し、AIが理解できるサイズに落とし込む技術。

Tools

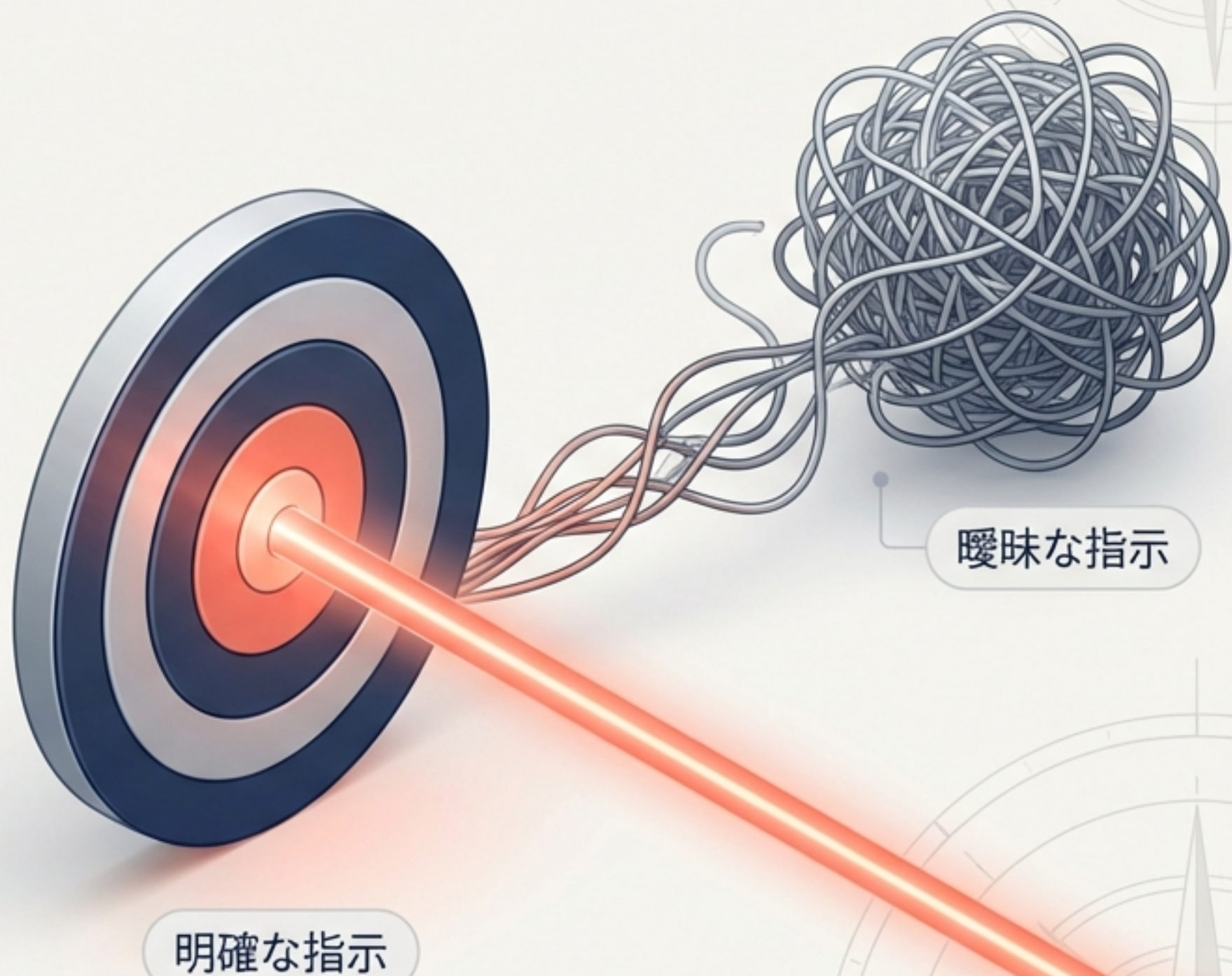
ファイルや仕様の要約の渡し方、`.cursor/rules` や `@` (コードベース参照) の活用。

Concept

「曖昧な指示」と「明確な指示」の出力結果の違いを体感する。

Deliverable

プロンプト改善前後の比較表、プロジェクト用ルールの作成 (1件)。



第3回：出力の検証とハルシネーション対策

Focus

生成されたコードの「意図」を確認し、テストし、自らの手で修正するプロセス。

Core Action

「実在しないAPI」を疑う視点を持つ。AIの出力を、公式ドキュメントや既存のコードベースと照合する習慣を身につける。

Deliverable

AI出力の検証チェックリスト、不適切な出力の修正実例。



第4回：実務への接続（レビューとリファクタリング）

Focus

AIを「レビュアー」や「リファクタリング提案者」として活用する高度なアプローチ。

Core Action

AIに複数のリファクタ案やテスト案を出させ、人間（あなた）がその採用可否を判断する。AIが生成したコードに対するレビュー観点を確立する。

Deliverable

リファクタリング前後のコード比較、AI生成コードに対するレビュー記録（1件）。



知識を実力に変える「4つのサイクル」

4. 締め (Commit)

次回までに「**現場で1つだけ
試す・実践する**」アクション
を宣言する。

3. 実践 (Practice)

【最重要】 実際に手を動か
し、プロンプト作成や検証、
ルール追加を行う。



1. 冒頭 (Share)

前回の実践内容や、効果的
だったプロンプトをチームで
短く共有。

2. 本題 (Learn)

その日のテーマの要点を、資
料とデモを通じてインプッ
ト。

本プログラムであなたが手にする成果物

プログラム終了時、あなたは単なる「知識」だけでなく、実務ですぐに使える具体的なツールキットを手にしていきます。



プロンプト対応メモ

どのような指示が最適なコードを生むかの実例集。



検証チェックリスト

ハルシネーションやセキュリティリスクを防ぐ防衛策。



Team Rules要約

.cursor/rules に組み込める、既存コードベースとの協調ルール。



レビュー記録

AIの出力を審査し、品質を担保した実績。

修了基準：AI開発者としての証明

- **実体験の完了**

AIツールを用いた開発から、生成結果の検証・修正までの一連のサイクルを1回以上、自力で完遂すること。

- **評価観点**

1. **プロンプトの適切さ**

タスクを正しく分解し、文脈を渡せているか。

2. **検証能力**

AIの出力を鵜呑みにせず、意図と正しさを検証できているか。

3. **セキュリティとルールの遵守**

機密情報の取り扱いや、ハルシネーション防止ルール（Team Rules等）と整合しているか。



セッション1に向けてのネクストステップ

Step 3: 「試したい課題」のピックアップ

普段の業務で「AIに任せてみたい」と思う小さなタスクや、リファクタリングしたいコードの断片を見つけておいてください。

Step 2: ポリシーの熟読

社内のAI利用ポリシー、および機密情報の取り扱いガイドラインを必ず一読してください。

Step 1: 開発環境のセットアップ

ご自身の環境でCursorをインストールし、ログインを完了させてください。



AIはエンジニアを奪わない。 AIを使いこなすエンジニアが未来を創る。

コーディングのスピードが劇的に上がる今、真に求められるのは「正しい方向へ舵を切る」能力です。

検証し、修正し、品質を担保する。

あなたはこの4回のセッションを通じて、単なるコードの書き手から、次世代の開発を牽引するキャプテンへと進化します。さあ、始めましょう。