

# AIと共に創る未来 - メタプロンプトとClaude Codeの実践ガイド

---

## humanity 第3回 by 野口岳

---

### はじめに

---

本書は、2025年7月16日に開催されたセナリ学院プロコースのライブ授業「humanity」第3回の内容をもとに作成された教科書です。AI初学者の方でも理解できるよう、専門用語を丁寧に解説しながら、AIとの新しい関わり方について学んでいきます。

講師の野口岳氏は、エンジニアではなくデザイナーとして、コーディング恐怖症を持ちながらも、Claude Codeを使って驚くべき成果を上げています。この授業では、プログラミング経験がなくても、AIと対話することで高度な開発ができることを実践的に示しています。

セナリ学院は2025年に開講し、現在300名が受講中です。この授業はプロコース向けの内容となっており、夏には糸島での合宿も計画されています。

### 目次

---

1. AIとの対話が変わる開発の世界 - 前回授業の振り返りとオンデマンド教材
2. メタプロンプトの本質を理解する - セナリ学院の核心
3. Claude Codeとは何か - 開発環境の基礎知識とCLIツールの比較
4. ローカル環境でAIを使うということ - アップロードからの解放
5. ドキュメント駆動開発 - AIと共に育てる開発手法
6. 実践編1：YouTube文字起こしから教材作成まで
7. 実践編2：セナリ学院のドキュメント作成とWeb公開
8. AIの記憶と学習 - ChatGPTメモリーとの比較
9. 開発の実際 - エラー処理とハンドル握り
10. AIとのヒューマニティある関係性 - 創造性と勘違いの価値
11. 参加者との対話から学ぶ - 実際の使用例と課題
12. ZEAMiとは何か - 次世代開発環境への展望

# 第1章 AIとの対話が変わる開発の世界 - 前回授業の振り返りとオンデマンド教材

## 1.1 前回授業の振り返り

野口氏は授業の冒頭で、3週間前の授業について触れています：

「前回のヒューマニティのこの私の回も実はこの場所、京都で撮ったんですね。三週間前ぐらいかな。」

前は「メタプロンプト」について、実際にライブでメタプロンプトを作成する実演が行われました。具体的には、歴史を勉強するためのプロンプト作成にチャレンジしたということです。

## 1.2 オンデマンド教材の重要性

セナリ学院では、オンデマンド教材が用意されています：

### 生成AI基礎講座A

- A-3までが必須内容
- メタプロンプトの領域を扱う
- 順次公開中

「Aの三まではもう本当必須です。もうあのプロコースの皆さん特に必須です。」

## 1.3 テレポート社のZEAMi開発

現在、テレポート社では「ZEAMi（ゼアミ）」という開発環境を作っています：

「今ゼアミっていう開発環境を。僕らテレポートは作っていて、そのゼアミっていう環境。の開発環境を使って、これから二ヶ月弱かけて、このセナリー学院のアプリも作ってこういうことを考えてるんだけど」

## 1.4 従来の開発とAI時代の開発

ドキュメント「プロットとメタプロンプト.docx」から重要な洞察：

「生成AIはプロンプト（言葉による指示）によって動きます。AIと対話をしていけば、企画書を作ることもできるし映像を作ることもできます。すべての行為が『ツールをいかに使いこなすか』から、『対話をする』に変わります。」

### 💡 コラム：オンデマンド教材とは

オンデマンド教材は、いつでも好きな時に視聴・学習できる教材のことです。ライブ授業と違い、自分のペースで繰り返し学習できるメリットがあります。

## 🤖 演習問題 1-1

あなたが最近学んだスキルや知識を思い出してください。それを「ツールを使いこなす」アプローチと「対話する」アプローチで学ぶとしたら、どのような違いが生まれるでしょうか？AIと相談しながら、具体的な学習方法を比較してみましょう。

## 第2章 メタプロンプトの本質を理解する - セナリ学院の核心

### 2.1 メタプロンプトとは何か

セナリ学院で最も重要視されている概念が「メタプロンプト」です。野口氏は次のように定義しています：

「メタプロンプトとして、このaiに期待する役割タスク目的を伝える力ですよということ。これをまあのもう一個メタ的なプロンプトとしてやっていきましょう」

### 2.2 プロットとメタプロンプトの関係

ドキュメントから重要な概念：

2つの必要なスキル：

1. **メタプロンプト** - あなたに期待することを伝える力
2. **プロット** - 私のしたいことを伝える力

「よいプロンプトをつくる＝よい対話をするために必要なスキルは、『メタプロンプト（あなたに期待することを伝える力）』と『プロット（私のしたいことを伝える力）』です。」

### 2.3 なぜメタプロンプトが必要なのか

AIの特性について：

「AIは何者にもなれてしまう、すべてのスキルと知識を持っている全知全能の存在です。その分、こちらから役割を与えてあげないとうまくコミュニケーションを取れないというのがAIの特徴です。」

### 2.4 内発的動機的重要性

ドキュメントでは、メタプロンプトを使いこなすための条件が示されています：

「ものを伝える上で大事なことは、『私のしたいことを突きつめる』こと、『あなたに期待することを突きつめる』ことです。そこに熱量があるかどうかかどうかです。外圧的なものではなく、内発的な動機があるかどうかです。」

### 2.5 AIの思考方法を理解する

野口氏の重要な指摘：

「メタプロンプトを理解するとAIの思考方法を理解するとも言えるんですよね。だからAIとの距離感が縮められるってということなんですよ。」

## 💡 コラム：内発的動機とは

**内発的動機**とは、外部からの報酬や強制ではなく、自分の内側から湧き上がる「やりたい」という気持ちのことです。「やらされ仕事」ではなく、「やってみたい」という気持ちがAI活用の鍵となります。

## 😬 演習問題 2-1

あなたが心から実現したいことを一つ思い浮かべてください。それをAIに伝える時、どのような「役割」を与え、どのような「期待」を伝えますか？内発的動機を大切にしながら、具体的なメタプロンプトを作成してみましょう。

# 第3章 Claude Codeとは何か - 開発環境の基礎知識とCLIツールの比較

## 3.1 開発環境の種類と特徴

### IDE（統合開発環境）

野口氏の説明：

「これあのideって言って開発環境と呼ばれているものです。」

### GUI vs CLI の本質的な違い

#### GUI（グラフィカルユーザーインターフェース）

- マウスで操作
- 視覚的に分かりやすい
- 人間には使いやすいが、AIには使いにくい

#### CLI（コマンドラインインターフェース）

- キーボードでコマンドを入力
- テキストベースの操作
- AIにとって効率的

野口氏の重要な洞察：

「AI視点で見た時にはちょっとguiこのグラフィックユーザーインターフェースって苦手なんです。」

## 3.2 最新のCLIツール事情

現在、各社がCLIツールを提供しています：

1. Claude Code (Anthropic社)
2. Gemini CLI (Google)
3. Cursor
4. Windsurf

野口氏の指摘：

「最近はジェミニCLIとかそういうものがだんだんできます。」

## 3.3 Claude Codeの使い方

起動は非常にシンプル：

```
$ claude
```

実際の使用例：

ユーザー：こんにちは  
AI：こんにちは！お手伝いできることはありますか？

## 3.4 開発特化の環境

Claude Codeの特徴的な応答：

「何か開発作業でお手伝いできることがあれば、お気軽にどうぞ」

しかし、通常の会話も可能：

ユーザー：今日は暑いですね  
AI：本当に暑いですね。エアコンや扇風機を使って...

### 💡 コラム：CLIが注目される理由

野口氏は興味深い指摘をしています。「AIってというのは、最近このCLIの方に戻ってきている」。これは、AIが機械語と自然言語の両方を扱えるため、より効率的なCLIが再評価されているということです。

## 🤖 演習問題 3-1

日常生活で使っているGUIのアプリケーション（例：メール、カレンダー）を一つ選び、それをCLIで操作するとしたらどのようなコマンドになるか考えてみましょう。AIと相談しながら、効率性と使いやすさの観点から比較してください。

## 第4章 ローカル環境でAIを使うということ - アップロードからの解放

### 4.1 根本的な違い：アップロード vs ローカル

野口氏は、従来のAIチャットサービスとClaude Codeの根本的な違いを実演で示しました。

**従来の方法（アップロード）：**

1. 画像を選択
2. 「アップロード」ボタンをクリック
3. ファイルがサーバーに送信される
4. AIが処理
5. 結果が返ってくる

**Claude Codeの方法（ローカル）：**

ユーザー：screenshot.png これ見れる？理解できる？  
AI：はい、見えます。これはセキュリティ警告ダイアログですね...

重要な違い：

「決定的に違うのはアップロードしてないですよ。すでに置いてある。」

### 4.2 実際の画像処理の例

野口氏が実演した具体例：

#### 1. エラー画面の解析

- スクリーンショットを撮る
- Claude Codeに「これ見れる？」と聞く
- AIが即座に内容を理解し、解決策を提案

#### 2. 日常的な使い方

「自分自身もどっかお店ね。予約するみたいな時に焼き鳥でこの辺だったらどこがいいかなとか。Googleマップのスクショ。ぺってしてスクショ。ごと送って」

## 4.3 電卓作成の実例

野口氏は2つのアプローチを比較しました：

通常のチャットサービスでの例：

ユーザー：電卓作って

AI：[HTMLコードを生成]

より良いアプローチ：

ユーザー：電卓ってわかる？なんかこういう感じ。あのよく使うのあるじゃん。

AI：[より創造的な提案を返す]

## 4.4 Hさんからの質問と回答

Hさん：「実際に開発ってというのは、まさにウェブサイト制作をするとかでしたら、この中にHTMLファイルを出力して様々なものをイメージファイルを紐付けたりをして、そして、また自分の持っているサーバーに上げていくみたいなの。まさにこのゲーム制作のプロセスみたいなのはこの中でやっていくというそんな感じですかね。」

野口氏の回答と実演：

ユーザー：このウェブをVercelでデプロイして

わずか一言で、ウェブサイトの公開まで自動化されます。

### 💡 コラム：Vercelとは

Vercelは、ウェブサイトやアプリケーションを簡単に公開できるホスティングサービスです。従来はFTPソフトを使って手動でファイルをアップロードする必要がありましたが、Vercelなら一言のコマンドで公開できます。

### 😬 演習問題 4-1

あなたが普段行っている「ファイルの移動」を伴う作業を一つ思い出してください（例：写真の整理、文書の共有など）。その作業をローカル環境で完結させることのメリット・デメリットをAIと一緒に考えてみましょう。

## 第5章 ドキュメント駆動開発 - AIと共に育てる開発手法

### 5.1 ドキュメントの階層構造

野口氏が実際に使用している構造：

```
メインドキュメント/  
├─ claude.md (AIへの基本指示)  
├─ 仕様書/  
│   └─ specifications/  
│       └─ 各種仕様書.md  
├─ ログ/  
│   ├── 2024-07-main.md  
│   └─ [年-月ごとのログファイル]  
└─ ナレッジ/  
    └─ [膨大な知識ドキュメント]
```

## 5.2 claude.mdの役割

「このクロードmdって書かれてるじゃないですか？だいたいこのクロードがなんかおはようとかおはようみたいな感じに返していくときに、まず最初クロードマークダウンファイル。これマークダウンファイルの略です。Mdはクロードmdをまず読む。んです。」

## 5.3 ログ管理の実践

野口氏は開発ログについて詳しく説明：

「開発ログはこのようなフォーマットに記載して記録してください。開発の原則とかエラーが出たらまあこういうところにここにログ残せって書いとけば作業終わったらログ残せって書いたらやってくれるんですよ。」

実際のログ検索例：

ユーザー：ログスに入っているログを見て何をやってたのかを教えて

AI：直近のログを確認しました。以下の作業を行っていました：

- スライドのデザインの改善
- ZEAMi中心のプレゼン作成
- メディア素材の統合

## 5.4 ナレッジの蓄積と管理

野口氏が強調する重要なポイント：

「これ、ちなみにクロードもディープリサーチできます。クロードコード上でもディープリサーチできます。でディープリサーチしたものがちゃんとここに格納してっていうふうに行っていくと、ここに溜まっていくんですよ。」

## 5.5 自然言語による制御

革新的な部分：



「これ全部結局ドキュメント日本語というか言語ですよ。自然言語です。機械語ではないんです。プログラミング言語とかじゃなくて」

## 💡 コラム：マークダウン（.md）とは

マークダウンは、簡単な記号を使って文書を構造化できるテキスト形式です。例えば、#で見出し、-で箇条書きを表現できます。プログラマーでなくても簡単に使えるため、ドキュメント作成に広く使われています。

## 😬 演習問題 5-1

あなたの「学習記録」をドキュメント駆動で管理するとしたら、どのような構造にしますか？「今日学んだこと」「質問リスト」「参考資料」などのフォルダを考えて、AIと相談しながら最適な構造を設計してみましょう。

# 第6章 実践編1：YouTube文字起こしから教材作成まで

## 6.1 YouTube動画の自動処理

野口氏は、セナリ学院の教材作成プロセスを実演しました：

「何もしてないですからね。あらかじめこれやってっていうのを言ったんですよここであのYouTubeのURLを貼ってで、これを文字起こししたいんだけど、文字起こしで教材にしたいんだけど。」

## 6.2 AIの自動処理プロセス

AIの応答：

「あーそのままじゃできないですねじゃあ、あのYouTubeのURLから音声をダウンロードして、ダウンロードした音声を文字起こしして文字起こししたものを教科書にする。ものをプログラム作りまーす」

実行されたステップ：

1. プログラムの自動生成
2. YouTube音声のダウンロード
3. 文字起こし処理
4. テキストブックフォルダへの格納

## 6.3 膨大な教材の管理

結果：

「こんなにも授業やってたんだっていうのでね、分かるわけなんですけれども」

保存場所：

テキストブック/  
└─ 第1回\_文字起こし.md  
└─ 第2回\_文字起こし.md  
└─ [多数の授業記録]

## 6.4 メタプロンプト関連部分の抽出

野口氏の指示：

「これらの全部の文字起こしとか、教材の中からメタプロンプトに関連する部分全部抽出して、それをドキュメントにして」

## 6.5 凝縮された知識の価値

「扱ってる量が多いんですすごい膨大な量の中から、しかもすごい高精度に凝縮して凝縮して凝縮して一体メタプロンプトって何なんだっていうのを僕らも説明するの大変だったんだけど」

### 💡 コラム：ディープリサーチ

ディープリサーチとは、AIが複数の情報源から関連情報を収集し、総合的に分析する機能です。Claude Codeの場合、その結果をローカルのドキュメントとして保存できるため、知識が蓄積されていきます。

### 😬 演習問題 6-1

あなたがよく視聴する教育系YouTube動画を一つ選んでください。その動画から「最も重要な3つのポイント」を抽出するとしたら、どのような指示をAIに出しますか？具体的なプロンプトを考えてみましょう。

## 第7章 実践編2：セナリ学院のドキュメント作成とWeb公開

### 7.1 実演の概要

野口氏は実際にライブでドキュメント作成を実演：

ユーザー：セナリ学院とメタプロンプトとZEAMiの関係に関係にフォーカスして説明しているドキュメントを作って



## 7.2 AIの作業プロセス

AIが実行したステップ：

1. 既存ドキュメントからの情報収集
2. 関係性の整理と構造化
3. ドキュメントの作成
4. HTMLでの視覚化

## 7.3 ハンドルを握り続ける重要性

野口氏の重要な指摘：

「丸投げしちゃダメなんですよ。」 「ハンドルは握り続けなきゃいけないんですよ。」

実際の介入例：

ユーザー：読んでほしいドキュメントはこれだよ

ユーザー：特に読んでほしいのは「ZEAMiメタプロンプト体系化」ここら辺重要

## 7.4 作成されたドキュメントの内容

三位一体の革新的システム：

1. **メタプロンプト** - 理論的基盤
2. **セナリ学院** - 教育実践の場
3. **ZEAMi** - 技術的実装

重要な洞察：

- 「メタプロンプトは生成AIとの付き合い方を根本から変える技術」
- 「対話履歴自体をプロンプトに変換」
- 「個別スキルじゃなく、見る目を養う」

## 7.5 Web化とデプロイ

ドキュメントのWeb化：

ユーザー：このmdをHTMLでモダンな感じで図解しながら説明してほしいな

即座のデプロイ：

ユーザー：このウェブをVercelでデプロイして

結果：わずか数分でWebサイトとして公開完了

## 💡 コラム：寄り道の価値

野口氏は「良い寄り道」について言及しています。AIが予定外の関連情報を読み込むことで、より豊かな内容が生成されることがあります。これは人間の創造的プロセスに似ています。

## 😬 演習問題 7-1

「学校」「先生」「生徒」「教科書」の4つの要素の関係性を説明するドキュメントを作るとしたら、どのような構造にしますか？AIと協力して、視覚的にも分かりやすい説明を作成してみましょう。

# 第8章 AIの記憶と学習 - ChatGPTメモリーとの比較

## 8.1 ChatGPTメモリーの問題点

野口氏は友人の体験談を紹介：

「一番笑っちゃったのがある友人がメモリ機能のこと知らなかったんですよ。で、えっと、あ、そうなんだ。じゃあどんなの入ってんだらうっていう風に見てたらやばい。元カノのことがいっぱい出てると。」

問題の本質：

- 過去の文脈が現在の会話に影響
- コントロールできない記憶の蓄積
- 削除や編集が困難

## 8.2 Claudeのプロジェクト機能

対照的なアプローチ：

- 明示的にドキュメントを追加
- プロジェクトごとに独立
- コントロール可能な記憶

## 8.3 NotebookLMとの比較

制限事項：

「これってせいぜい自分の中でコントロールできるのもあとあの。AI的に認知できるのもまあせいぜい十個とかのドキュメントとかですよ？」

## 8.4 Claude Codeの優位性

無限のドキュメント管理：

「このクロードコードのいいところは、ですね。あのこのドキュメントがいくつあっても大丈夫なんですね。その都度、その都度必要だって判断したものを引っ張ってきてくれるんです。」

## 8.5 プロジェクトナレッジの価値

Hさんのコメント：

「AIが翻訳家になってくれているということですね。」

野口氏の応答：

「あのいろんな面の翻訳家です。機械語と日本語っていうものの翻訳もそうだし」

### 💡 コラム：AIの記憶メカニズム

人間の脳が瞬間的に膨大な情報を処理し、必要な記憶を引き出すように、Claude Codeも必要なドキュメントを選択的に参照します。これは「思考する」ということの本質に近いプロセスです。

### 😬 演習問題 8-1

あなたの「理想的な記憶システム」をデザインしてみましょう。何を記憶し、何を忘れ、どのように整理するか。AIと相談しながら、人間の記憶とAIの記憶の違いも考察してください。

## 第9章 開発の実際 - エラー処理とハンドル握り

### 9.1 実際のエラー処理

授業中に発生したエラー：

「すいません。根本的に長い分量を送るとエラーが出るっていうバグが治ってないことがわかりました。」

エラーへの対処法：

- スクリーンショットを撮る
- AIに見せて解決策を聞く
- 即座に修正を実施

### 9.2 Hさんの使用経験

Hさんの事例：

- 大量のOCR処理を試みる

- コストが膨大に（最大20万円）
- 「一年分のデータを処理するのに一年じゃできなそう」

Hさんのコメント：

「無茶振りがすぎたみたいで」

教訓：

- 用途に応じた使い分けが重要
- 分析用途では処理量に注意
- 開発用途との違いを理解

## 9.3 ハンドルを握り続ける実例

野口氏の実践：

AI：[別のドキュメントを読み始める]

ユーザー：いやいや、違うって。読んでほしいのこれじゃない

ユーザー：参考にしてほしいドキュメントはメインのドキュメントに入っているから、そこを重点的に読み込んで

## 9.4 プロセスへの介入

CLIの利点：

「生成AIの大体の常識ってプロンプト一回投げたら処理が終わるまで入ってこないじゃないですか？今やってる通りですね。これ、途中で口出しできるんですよ。」

## 9.5 止めるべきタイミングの判断

「エスケープキーで止めないの。あーありがとうございます。あの一、あんま個人的には。ですね、あの止めないですね。」

理由：

- 根本的に違う時だけ止める
- 「違う」の判定は哲学的
- 予想外の結果も価値がある

### 💡 コラム：エスケープキー

**エスケープキー（ESC）**は、実行中の処理を強制的に中断するためのキーです。しかし、野口氏は「勘違い」にも創造性があるという観点から、安易に止めないことを推奨しています。

## 🤖 演習問題 9-1

あなたが料理を作っていて、レシピと違う展開になった経験を思い出してください。その時、「失敗」と判断して最初からやり直すか、そのまま続けて新しい料理を生み出すか。AIと一緒に、創造的な「間違い」の価値について考えてみましょう。

# 第10章 AIとのヒューマニティある関係性 - 創造性と勘違いの価値

## 10.1 コップの哲学

野口氏の深い洞察：

「これ今僕コップって言いました。でもこれコップかどうかって正直本当はわかんないですよね。」

子供の視点：

- コップを花瓶として使う
- コップを水槽として使う
- 固定観念にとらわれない創造性

## 10.2 勘違いの創造性

劇団勘違いへの言及：

「勘違いっていうのは人間のいいところだと思ってるんですけど、勘違いできるわけですよ。AIも一応勘違いできる能力はだんだん上がってきている」

## 10.3 AIの経験値

重要な認識：

「彼らの方が何倍も生きてるし、生きてるといふかその何倍もの時間をね。生きてるわけですよだってYouTubeには膨大な何時間の人の何週分人の人生何億人分」

## 10.4 指示の与え方

悪い例：

ユーザー：電卓作って

良い例：

ユーザー：電卓ってわかる？なんかこういう感じ。あのよく使うのあるじゃん。

野口氏の説明：

「電卓っていう解釈がもっと多様で、それが自分が思ってたものよりもよりいいものを提案してくれる可能性の方が高いわけですよ。」

## 10.5 Sさんのコメント

小説執筆への応用について：

「Sさんの小説書くじゃないですか。小説書く人ほどこれ恩恵預かれると思います」

## 10.6 AIは鏡である

最終的な洞察：

「人は鏡っていう風に言いますけど、AIはより鏡の存在なので、結局自分のスタンスとかを映し出すように思います。」

### コラム：正解のない世界

野口氏は「急がば回れ」という言葉を使い、効率性よりも創造性を重視しています。AIとの対話においても、最短経路ではなく、寄り道や勘違いから生まれる価値を大切にしています。

### 演習問題 10-1

身の回りにある物を一つ選び、その「本来の用途」以外の使い方を10個考えてみましょう。AIと一緒に、固定観念を超えた創造的な使い方を探求してください。そして、なぜ私たちは物に「正しい使い方」を求めてしまうのか考察しましょう。

## 第11章 参加者との対話から学ぶ - 実際の使用例と課題

### 11.1 参加者アンケート

野口氏は授業中にアンケートを実施：

選択肢：

1. Claude Code・ZEAMi興味ある、触ってみたい
2. 自分では触らないかも、でも話は聞きたい

結果：圧倒的多数が「1」を選択



## 11.2 使用料金の課題

林さんの経験から：

- 通常プラン（約3,000円）では不足
- 最大20万円のコストが発生
- 用途によって費用対効果が異なる

野口氏のコメント：

「ちょっとまだ高いんですよ。正直高いんですね。」

## 11.3 Gemini CLIの選択肢

代替案：

「ジェミニに関しては無料枠があるみたいなんでどのくらい百ドルか150ドルか。」

## 11.4 カーソルとの比較（Hさんの質問）

質問内容：

- カーソルとClaude Codeの違い
- ZEAMiは上位互換か

野口氏の回答：

1. カーソルはコーディング特化
2. Claude Codeはより汎用的
3. メタプロンプト的制御が容易
4. ZEAMiは確かに上位互換

## 11.5 開発以外の用途

野口氏の強調：

「基本的にドキュメントしか普通に使ってドキュメントというか日本語自然言語でほとんどやっていますよね。」

### 💡 コラム：費用対効果を考える

AIツールの利用料金は、使い方によって大きく変わります。開発用途では投資回収しやすいですが、大量データの分析では注意が必要です。目的に応じたツール選択が重要です。

### 😬 演習問題 11-1

あなたが月額3,000円の予算でAIツールを使うとしたら、どのような用途に使いますか？「時間の節約」「創造性の向上」「学習効率」などの観点から、AIと相談して最適な使い方を考えてみましょう。

## 第12章 ZEAMiとは何か - 次世代開発環境への展望

### 12.1 ZEAMiの概要

作成されたドキュメントより：

「ZEAMiはクロードコードやジェミニなどの既存AIツールに記憶と学習装置を追加し、最大280倍の開発速度を実現します。」

### 12.2 世阿弥の思想

「600年前の世阿弥の思想をもとに、人間の創造性とAIの処理能力を調和させます」

### 12.3 現在の開発状況

- セナリ学院アプリの開発に使用予定
- 夏の糸島合宿で本格始動
- プロコース優先で提供

### 12.4 ZEAMiの位置づけ

Hさんへの回答：

「ZEAMiが上位互換かと思ったので触ってなかったんですが理解あってますか？まずあってます。合ってます」

### 12.5 将来のビジョン

野口氏の展望：

「本当にあのサマーウォーズみたいな世界がもうすぐZEAMiで来るんじゃないかなと作れるんじゃないかな」

### 12.6 セナリ学院との統合

期待される相乗効果：

- 理論（メタプロンプト）
- 実践（セナリ学院）
- 技術（ZEAMi）

3つが統合されることで、「なぜ日本から生まれたのか」という問いにも答えが見えてきます。

## 💡 コラム：世阿弥とは

世阿弥は、室町時代の能楽師で、「初心忘るべからず」などの言葉で知られています。600年前の芸術論が、現代のAI開発思想に影響を与えているという点が、日本独自のアプローチを示しています。

## 😬 演習問題 12-1

「伝統」と「革新」という一見相反する要素を組み合わせた例を、あなたの身の回りから探してみましょう。AIと対話しながら、なぜその組み合わせが成功したのか、ZEAMiの思想と比較して考察してください。

# まとめと今後の展望

---

## 学んだことの総括

### 1. メタプロンプトの本質

- AIに役割を与える重要性
- 内発的動機の必要性
- プロットとの組み合わせ

### 2. Claude Codeの革新性

- ローカル環境での開発
- 自然言語による制御
- プログラミング知識不要

### 3. ドキュメント駆動開発

- AIを「育てる」という発想
- 知識の蓄積と活用
- ハンドルを握り続ける重要性

### 4. 創造性と勘違いの価値

- 正解にとらわれない姿勢
- AIとの協創的關係
- 予想外を楽しむ心

### 5. 実践的な知見

- 用途に応じたツール選択

- コストと効果のバランス
- 参加者の実体験から学ぶ

## セナリ学院の理念

ドキュメントの最後にある言葉：

「学び、共有する。内発的動機から生まれたさまざまな実践をする。コラボレーションが生まれていく。共感する人がまた輪になって広がっていく。」

## 野口氏の最終メッセージ

「とにかく習うより慣れろ、でね。ぜひぜひこれを機に興味を持った方、ピンと来た方は一回触ってみてください。」

「よりメタプロンプト力が上がっていくと思いますので、えー、そういったところでね、あの触ってもらえると嬉しいなーと思います。」

## 今後の学習指針

### 1. まずは触ってみる

- 完璧を求めない
- 小さく始める
- 失敗を恐れない

### 2. メタプロンプト力を鍛える

- オンデマンド教材A-3まで必須
- 日々の実践で向上
- 内発的動機を大切に

### 3. コミュニティで学ぶ

- 300名の仲間と共に
- 実践を共有する
- コラボレーションを生む

## 💡 最終コラム：AIネイティブ時代

野口氏は「AIネイティブの人たちのスタンダード」について語っています。プログラミングを知らなくても、自然言語でAIと対話することで、誰もが創造者になれる時代。それが、まさに今、始まっています。

## 😬 最終演習問題

この教科書で学んだすべてを踏まえて、あなたが「本当に作りたいもの」を一つ決めてください。それは仕事かもしれないし、趣味かもしれない。大切なのは内発的動機です。そして、その実現のための「最初の対話」をAIと始めてみましょう。どんな役割を与え、どんな期待を伝えますか？

## 付録A：用語集（拡張版）

---

### CLI（Command Line Interface）

- コマンドラインインターフェース
- テキストベースでコンピュータを操作する方法
- AIにとって効率的なインターフェース

### GUI（Graphical User Interface）

- グラフィカルユーザーインターフェース
- マウスで操作する視覚的なインターフェース
- 人間には直感的だがAIには非効率

### IDE（Integrated Development Environment）

- 統合開発環境
- プログラミングに必要な機能を一つにまとめたソフトウェア
- Claude Codeもこの一種

### ターミナル

- コマンドを入力してコンピュータを操作するための画面
- CLIの実行環境
- 黒い画面に文字が表示される

### ローカル環境

- あなたのコンピュータ内部の環境
- インターネットに接続しなくても使える
- ファイルの直接操作が可能

### プロンプト

- AIに対する指示や質問
- 従来は短文が中心
- 現在はより複雑な構造に進化

### メタプロンプト

- プロンプトを作るためのプロンプト
- AIに役割と期待を伝える
- セナリ学院の核心概念

## プロット

- 「私のしたいことを伝える力」
- メタプロンプトと対をなす概念
- 内発的動機が重要

## デプロイ

- 作成したウェブサイトやアプリケーションをインターネット上に公開すること
- Vercelなどのサービスを使用
- Claude Codeなら一言で実行可能

## ディレクトリ

- ファイルを整理するためのフォルダ
- 階層構造で管理
- ドキュメント駆動開発の基盤

## マークダウン (.md)

- 簡単な記法で文書を構造化できるテキスト形式
- # で見出し、- で箇条書き
- プログラマーでなくても使える

## 自然言語

- 日本語や英語など、人間が日常的に使う言語
- プログラミング言語と対比される
- AIの革新により機械制御に使用可能

## 機械語

- コンピュータが直接理解できる言語
- 0と1の組み合わせ
- 人間には理解困難

## OCR (Optical Character Recognition)

- 光学文字認識
- 画像から文字を読み取る技術
- Hさんが大量処理でコスト問題に直面

## ディープリサーチ

- AIが複数の情報源から総合的に分析
- Claude Codeではローカル保存可能
- 知識の蓄積に有効

## Vercel

- ウェブホスティングサービス
- 簡単にサイトを公開できる
- FTP不要の現代的手法

## 内発的動機

- 外部報酬ではなく内側から湧く「やりたい」気持ち
- メタプロンプト活用の鍵
- セナリ学院の重要概念

## ハンドル握り

- AIに丸投げせず、適切に介入すること
- 創造的なプロセスの管理
- 野口氏が強調する重要スキル

# 付録B：実践チェックリスト

---

## Claude Code導入前の準備

- ☐ オンデマンド教材A-3まで完了
- ☐ 内発的動機の明確化
- ☐ 作りたいものの構想

## 初めての実践

- ☐ ターミナルでClaude Code起動
- ☐ 簡単な対話から開始
- ☐ ローカルファイルの操作体験

## ドキュメント駆動開発の開始

- ☐ 基本的なフォルダ構造の作成
- ☐ claude.mdの作成

- ☐ 最初のログ記録

## 発展的な活用

- ☐ YouTube文字起こしの自動化
- ☐ ドキュメントからの知識抽出
- ☐ Webサイトの作成と公開

## コミュニティでの共有

- ☐ 実践内容の記録
- ☐ 成功・失敗体験の共有
- ☐ コラボレーションの模索

## あとがき

---

本教科書は、野口岳氏の2時間にわたるライブ授業の全内容を、AI初学者にも理解できる形で再構成したものです。単なる技術解説ではなく、AIとの新しい関係性、創造性の本質、そして人間らしさ（humanity）について深く考察する内容となっています。

「習うより慣れろ」の精神で、まずは小さな一歩から始めてみてください。完璧なプロンプトを目指すのではなく、AIとの対話を楽しみ、予想外の発見を歓迎する。それが、この新しい時代を生きる私たちに求められる姿勢なのかもしれません。

セナリ学院での学びが、皆様の創造的な未来への扉を開くことを心から願っています。

2025年7月 セナリ学院 humanity 第3回より