株式会社インフォメーション・ディベロプメント様向け

ハンズオンガイド

Day2 CLAUDE.md、ハーネス、コンテキスト

講師はハーネスなし、受講者はハーネスありで同じ依頼を送り、画面の差を見比べます。CLAUDE.md、Subagents、Skills、Hooks、コンテキストの5つを順に確かめます。

2026年11月18日（水） Zoom 個人演習 Claude Code（WSLのUbuntu）

目次

本日のゴール

当日の流れ
- Zoomでの進め方

始める前の準備
- 配布フォルダの展開
- Claude Codeの起動
- モードの使い分け
- 見比べの進め方

CLAUDE.mdとAGENTS.md

D2-1 /initとAGENTS.mdの見比べ
- ねらい
- できあがりの目安
- 使うファイルと作るファイル
- STEP 1 AGENTS.mdの確かめ
- STEP 2 /initでのCLAUDE.mdづくり
- STEP 3 2つのファイルの見比べ
- うまくいかないとき
- 早く終わった方へ

ハーネスの4種類

D2-2 Subagentsの見比べ
- ねらい
- できあがりの目安
- 使うファイルと作るファイル
- STEP 1 予想
- STEP 2 同時に送る
- STEP 3 見比べ
- STEP 4 現場での置き換え
- うまくいかないとき
- 早く終わった方へ

D2-3 Skillsの見比べ
- ねらい
- できあがりの目安
- 使うファイルと作るファイル
- STEP 1 予想
- STEP 2 同時に送る
- STEP 3 見比べ
- STEP 4 現場での置き換え
- うまくいかないとき
- 早く終わった方へ

D2-4 Hooksの見比べ
- ねらい
- できあがりの目安
- 使うファイルと作るファイル
- STEP 1 予想
- STEP 2 同時に送る
- STEP 3 見比べ
- STEP 4 現場での置き換え
- うまくいかないとき
- 早く終わった方へ

MCPのデモ

コンテキストエンジニアリング

D2-5 コンテキストの見比べ
- ねらい
- 押さえること
- できあがりの目安
- 使うファイルと作るファイル
- STEP 1 CLAUDE.mdの参照と予想
- STEP 2 同時に送る
- STEP 3 見比べ
- STEP 4 現場での置き換え
- うまくいかないとき
- 早く終わった方へ

ハーネスの使い分け

よくある質問

01

本日のゴール

今日は、同じ依頼を、講師はハーネスの無い画面で、受講者はハーネスのある画面で送り、結果を見比べます。差を見つけたら、差を生んだファイルの名前まで戻ります。

本日の学び

- CLAUDE.mdとAGENTS.mdの役割の違いを知り、/initでCLAUDE.mdを作る
- Subagents、Skills、Hooksが、同じ依頼の結果をどう変えるかを自分の画面で見る

- AIに常に守らせる資料と、作業のたびに渡す資料を分けて置く

学びを確かめる成果物

- ☐ `見比べ/` の見比べ表5枚に、差と、差を生んだファイルの名前が書いてある
- ☐ 自分で育てた `CLAUDE.md` に、`@AGENTS.md` の行と、コンテキストの資料への参照の3行が入っている

題材は、サンプル情報サービス株式会社（架空）の勤怠・工数管理システムKINTAIです。演習では画面を動かさず、Claude Codeの応答と、書き換えたファイルで見比べます。画面を動かすのは、講師のデモと発展だけです。

02

当日の流れ

全体で [180min]です。休憩は前半のあとに [10min]あります。

枠	中身	形式	時間
開始	Day1 のおさらいと、Day2の配布フォルダの展開	説明	[10min]
講義	CLAUDE.mdと/init、AGENTS.mdの違い	講義	[10min]
D2-1	/initとAGENTS.mdの見比べ	個人演習	[10min]
講義	ハーネスとは。4種類（Subagents、Skills、Hooks、MCP）の地図	講義	[10min]
D2-2	Subagentsの見比べ	個人演習	[20min]
D2-3	Skillsの見比べ	個人演習	[20min]
休憩			[10min]
D2-4	Hooksの見比べ	個人演習	[20min]
デモ	MCP（講師の画面）	デモ	[10min]
講義	コンテキストエンジニアリング。AIに渡す資料の種類と置き場所	講義	[15min]
D2-5	コンテキストの見比べ	個人演習	[25min]
閉会	まとめ、質疑応答、アンケート	質疑	[20min]

Zoomでの進め方

- 講師がZoomで画面を共有し、同じ操作を見せながら進めます。手元で使うのは、WSLにつないだVS Codeと、その中のターミナルで動かすClaude Code、Windowsのブラウザです。画面が1枚の方は、Zoomの画面を小さくして横に並べてください。
- 演習はすべて個人で進めます。発表や、ほかの受講者との話し合いはありません。
- 質問や、詰まったときは、Zoomのチャットに「D1-2 STEP 3ブラウザに表が出ない」のように、演習の番号、STEP、起きていることを書いてください。講師が順に答えます。
- 各STEPの[Nmin]は目安です。時間になったら講師が次のSTEPへ進めます。結果が途中でも、次のSTEPへ移って構いません。

- 見出しに 発展 のバッジが付いたSTEPは、演習の枠の時間に入っていません。必須のSTEPが終わったら、発展を待たずに次の演習へ進みます。
- Claudeの応答を待つ間は、次のSTEPの本文を読んでください。
- 研修用の組織の利用枠を、受講者全員で使います。同じ依頼を何度も送り直さず、うまくいかないときはZoomのチャットで知らせてください。研修の時間の外で長く使い続けることも控えてください。

03

始める前の準備

配布ZIPの`idcc-day2_20261118.zip`をWSLのUbuntu-22.04で展開し、VS Codeで開いてClaude Codeを起動します。開始の [10min]の中で、全員の起動をそろえます。このPCには`Ubuntu`という別の環境も入っていますが、研修で使うのは`Ubuntu-22.04`だけです。`Ubuntu`の方では、Claude Codeが使えず、配布フォルダも見えません。

配布フォルダの展開

1. 教材サイトのDay2の欄から`idcc-day2_20261118.zip`をダウンロードします。Windowsの`ダウンロード`フォルダに保存されます。
2. VS CodeをWSLにつなぎ、`ターミナル` → `新しいターミナル`でターミナルを開きます。左下に`WSL: Ubuntu-22.04`と出ていれば、つながっています。`WSL: Ubuntu`と出ていたら別の環境です。接続を閉じて、`Ubuntu-22.04`を選び直してください。
3. ターミナルで`echo $WSL_DISTRO_NAME`と打ちます。`Ubuntu-22.04`と返れば、今いる環境が合っています。
4. 次のコマンドで、ZIPをホームにコピーして展開します。

ターミナルに打つコマンド

```
cd ~
cp /mnt/c/Users/*/Downloads/idcc-day2_20261118.zip ~/
unzip -o idcc-day2_20261118.zip
ls -a idcc-day2
```

`.claude`、`AGENTS.md`、`docs`、`src`、`見比べ`が並べば、展開できています。`CLAUDE.md`はまだありません。D2-1で自分で作ります。

ZIPはD2-4で設計書を配布の中身に戻すときに使うので、ホームに残しておいてください。消してしまったときは、`cp`の行から打ち直します。

Windowsのエクスプローラーからの確かめ方は、次のとおりです。アドレス欄に`\\wsl.localhost\Ubuntu-22.04\home\ユーザー名`と打つと、Ubuntu22.04のホームが開き、`idcc-day2`が見えます。配布フォルダの置き場所は、このホームです。

Claude Codeの起動

1. VS Codeの `ファイル` → `フォルダーを開く` で `/home/ユーザー名/idcc-day2` を開きます。

2. ターミナルで次のコマンドを打ちます。

ターミナルに打つコマンド

```
cd ~/idcc-day2
claude update
claude
```

`claude update` は、Claude Codeを最新版にするコマンドです。`Claude Code is up to date` か `Successfully updated` で始まる行が出れば済みです。ほかの表示やエラーが出たときは、そのまま先へ進み、講師に知らせてください。

1. このフォルダを信頼するかを聞かれたら、信頼する側の選択肢を選んでEnterを押します。

2. 入力欄の下に `manual mode on` と出ていることを確かめます。出ていなければ、出るまでShift+Tabを押します。

通常のモード（manual mode on）では、Claudeがファイルを書き換える前とコマンドを動かす前に、承認の画面が出ます。

配布フォルダの `.claude/settings.json` で、起動した直後のモードをmanualに決めてあります。この設定が無いフォルダでは、Claude Codeは `auto mode on` で起動するので注意してください。

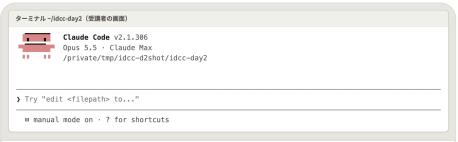

図1：起動した直後の画面。入力欄の下に `manual mode on` と出ている。プランの名前は契約によって変わる

モードの使い分け

Shift+Tabを押すたびに、入力欄の下の表示が `manual mode on` 、`accept edits on` 、`plan mode on` 、`auto mode on` の順に変わり、`manual mode on` に戻ります。研修ではauto modeを使いません。

演習	モード	入力欄の下の表示
D2-1、D2-2	通常のモード	`manual mode on`
D2-3、D2-4、D2-5	編集の自動承認	`accept edits on`

`accept edits on` では、ファイルの作成と書き換えに加えて、作業フォルダの中のmkdir、touch、cp、mv、rm、sedのコマンドも承認なしで進みます。それ以外のコマンドと `.claude/` の中への書き込みは、承認の画面が出ます。

D2-3 からD2-5では、Claudeの応答に `rm` や `mv` の行が出たら、どのファイルを動かしたかを読んでください。承認の画面が出たときは、コマンドの全文を読んでから選びます。

見比べの進め方

D2-2 からD2-5は、同じ型で進めます。講師がZoomで共有するのは、ハーネスの無いフォルダの画面です。受講者は、ハーネスのあるフォルダで同じ文面を送ります。

STEP	中身	時間
STEP 1 予想	Claude Codeを起動し直す。送る文面と効くはずのファイルを読み、何が変わりそうかを見比べ表に1行で書く	[3min]
STEP 2 同時に送る	講師の合図で、講師と同じ文面を送る。待つ間に、効くはずのファイルの中身を読む	[8min]
STEP 3 見比べ	講師の画面と自分の画面の差を2つ、見比べ表に書く。差を生んだファイルの名前も書く	[6min]
STEP 4 現場での置き換え	自分の現場なら、このハーネスに何を書くかを見比べ表に1行で書く	[3min]

D2-5 だけは、STEP 1でCLAUDE.mdに資料への参照を3行足すので、STEP 1が [8min]です。

見比べの前の起動し直し

STEP 1 の初めに、講師も受講者も `/exit` でClaude Codeを終え、 `claude` で起動し直します。前の演習の会話を持ち越すと、 `/context` の数字を講師の画面と比べられないためです。CLAUDE.mdは起動したときに読み込まれるので、書き換えたあとも起動し直します。会話を空にする `/clear` は、見比べの前には使いません。

見比べ表は `見比べ/` に5枚あります。VS Codeで開き、表の空いた欄に書き込んでCtrl+Sで保存してください。

> **同じ文面は1回だけ送ります。**研修用の組織の利用枠を、全員で使っています。うまくいかないときは送り直さずに、Zoomのチャットで知らせてください。出力は毎回ばらつくので、講師の画面と同じ差が出ないこともあります。そのときは、差が出た方の画面を講師が言葉で紹介します。

> **Tips** Zoomの画面共有を見る間は、VS Codeを最小化するとZoomが止まりにくくなります。 `/context` の表示の前は、特にそうしてください。

04

CLAUDE.mdとAGENTS.md

ファイルとコマンド	読む、使うのは	書くこと
`AGENTS.md`	Claude Codeを含む、複数のAIのツール	どのツールにも守らせる決まり。応答の長さ、触らないフォルダ、題材の説明
`CLAUDE.md`	Claude Code	Claude Codeに伝えること。コマンド、層の分け方、ハーネスの置き場所
`/init`	人が入力欄に打つ	Claudeにフォルダを読ませ、CLAUDE.mdの下書きを作らせる

Claude Codeは、作業フォルダとその上のフォルダにCLAUDE.mdが無いときに、代わりにAGENTS.mdを読みます。CLAUDE.mdがあると、Claude CodeはCLAUDE.mdだけを読み、AGENTS.mdは直接は読みません。

両方を効かせるには、CLAUDE.mdの先頭に`@AGENTS.md`の1行を書いて取り込みます。`@ファイル名`の行は、そのファイルの中身をCLAUDE.mdの中に書いたのと同じ扱いにします。D2-5でコンテキストの資料を取り込むときも、同じ書き方です。

読み込んでいるファイルは、入力欄に`/memory`と打つと一覧で出ます。取り込んだファイルは`@-imported`と並びます。

出典は、Claude Codeの公式ドキュメントのmemoryのページにあるAGENTS.mdの節です（https://code.claude.com/docs/en/memory）。

05

D2-1 /initとAGENTS.mdの見比べ

[10min] 個人演習

この演習の [10min]は、必須のSTEP 1〜3の合計です。

ねらい

対象は、KINTAIのプロジェクト直下の指示ファイルです。/initでCLAUDE.mdを作り、最初から置いてあるAGENTS.mdと役割を見比べます。見比べは、作られたCLAUDE.mdの中身そのもので行います。CLAUDE.mdを作るとAGENTS.mdは直接は読まれなくなるので、取り込む1行が先頭に入っているかも確かめてください。

できあがりの目安

- ☐ `CLAUDE.md` ができていて、先頭に `@AGENTS.md` の行が入っている
- ☐ `見比べ/D2-1_CLAUDE.mdとAGENTS.md.md` に、2つのファイルの役割の違いが1行ずつ書いてある
- ☐ 起動し直したあとの `/memory` に、AGENTS.mdが `@-imported` で並んでいる

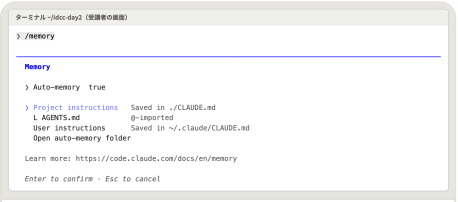

図2：できあがりの例。起動し直してから `/memory` を打つと、Project instructionsの下にAGENTS.mdが並ぶ

使うファイルと作るファイル

区分	ファイル	使い方
使う	`AGENTS.md`	STEP 1 で読む
作る	`CLAUDE.md`	STEP 2 で/initに作らせる
作る	`見比べ/D2-1_CLAUDE.mdとAGENTS.md.md`	STEP 3 で書き込む

AGENTS.mdを読む → /initでCLAUDE.mdを作る → 2つを見比べる

この演習は、起動したときの会話でそのまま進めます。最後にClaude Codeを起動し直し、作ったCLAUDE.mdを読み込ませます。

STEP 1 AGENTS.mdの確かめ

依頼を送らない [2min]

VS Codeで `AGENTS.md` を開いて読みます。どのツールにも守らせたい決まりを、短く書いたファイルです。

Claude Codeの入力欄に `/memory` と打ってEnterを押すと、いま読み込んでいる指示のファイルが出ます。見終わったらEscで閉じます。

入力欄に打つコマンド

```
/memory
```



図3： `/memory` の表示。CLAUDE.mdがまだ無いので、AGENTS.mdが読み込まれている

STEP 2 /initでのCLAUDE.mdづくり

[5min]

入力欄の下が `manual mode on` になっていることを確かめてから打ちます。

入力欄に打つコマンド

```
/init
```

Claudeがフォルダの中のファイルを読んでから、CLAUDE.mdを書きます。読むためのコマンドの承認が出たら、コマンドを読んでから1. Yesを選んでください。CLAUDE.mdを作る承認の画面では、ファイル名が `CLAUDE.md` になっていることを見てから1. Yesを選んでください。



図4：/initが終わったところ。CLAUDE.mdに何を書いたかが短くまとまって返った

STEP 3 2つのファイルの見比べ

依頼を送らない [3min]

VS Codeで `CLAUDE.md` を開き、先頭に `@AGENTS.md` の行があるかを見ます。無ければ、1行目の見出しのすぐ下に `@AGENTS.md` と書き足して保存してください。

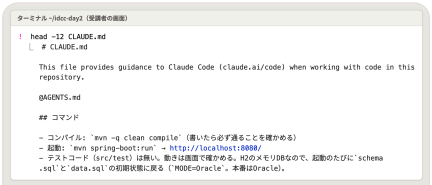

図5：/initが作ったCLAUDE.mdの先頭の例。見出しと説明の下に `@AGENTS.md` の行がある

`見比べ/D2-1_CLAUDE.mdとAGENTS.md.md` を開き、2つのファイルに何が書いてあるかを1行ずつ書きます。

書き終えたら、入力欄に `/exit` と打ってClaude Codeを終え、ターミナルで `claude` と打って起動し直します。起動し直してから `/memory` を打ち、AGENTS.mdが `@-imported` で並んでいれば完了です。

うまくいかないとき

/initが長く止まっているように見える

フォルダの中のファイルを順に読んでいます。応答の途中にファイル名が出ていれば進んでいます。[3min]を過ぎても何も出ないときは、Zoomのチャットで知らせてください。

CLAUDE.mdに `@AGENTS.md` の行が無い

VS Codeで `CLAUDE.md` の1行目の見出しのすぐ下に `@AGENTS.md` と書き足し、Ctrl+Sで保存してください。書き足したあとにClaude Codeを起動し直すと効きます。

`/memory`にAGENTS.mdが並ばない

`@AGENTS.md`の行の頭が`@`になっているか、行をバッククォートで囲んでいないかを見てください。直したら、Claude Codeを起動し直します。

CLAUDE.mdの中身が講師の画面と違う

/initが書く中身は、毎回少しずつ違います。`@AGENTS.md`の行があれば、ほかの中身が違っても演習は進められます。

早く終わった方へ

追加と考察

/initがCLAUDE.mdに書いたことのうち、AGENTS.mdと重なっている行と、AGENTS.mdに無い行を探します。AGENTS.mdに無い行は、どのファイルから読み取ったものかを考えて、見比べ表の下に1行で書いてください。

06

ハーネスの4種類

ハーネスは、依頼の文面の外からAIの振る舞いを決める仕組みです。どれも`.claude/`の下にファイルで置くので、同じフォルダを開いた全員に同じように効きます。

種類	置き場所	いつ効くか	Day2 の演習
CLAUDE.md	`CLAUDE.md`	常に読まれる	D2-1、D2-5
Subagents	`.claude/agents/`	依頼が説明に合うと、別のコンテキストで動く	D2-2
Skills	`.claude/skills/`	依頼が説明に合うと、手順が読み込まれる	D2-3
Hooks	`.claude/settings.json`と`.claude/hooks/`	ファイルの書き換えやコマンドの前後に、必ず動く	D2-4
MCP	`.mcp.json`	外のデータや道具を、Claudeの道具として使えるようにする	講師のデモ

SubagentとSkillは、ファイルの先頭に書いた説明（description）をClaudeが読み、依頼に合うと判断したときに使われます。HookはClaudeの判断を通らずに、決めた場面で必ず動きます。

07

D2-2 Subagentsの見比べ

[20min] 個人演習

この演習の [20min]は、必須のSTEP 1〜4の合計です。

ねらい

対象は、KIN030工数入力の実装と、docsの画面設計書KIN030の突き合わせレビューです。同じ文面を講師の画面（ハーネスなし）と自分の画面（ハーネスあり）で同時に送り、結果を書き出したファイル同士で見比べます。Subagentに分けると、親のコンテキストの使い方と結果の形がどう変わるかを見ます。

できあがりの目安

- ☐ `成果物/D2-2_レビュー結果.md` ができていて、見出しが3つ（指摘の件数、重要な不一致、見た観点）そろっている
- ☐ `見比べ/D2-2_Subagents.md` に、差が2つと、効いたファイルの名前が書いてある
- ☐ 自分と講師の `/context` のMessagesの数字を、見比べ表に書き写してある
- ☐ 自分の現場なら何を書くかが、1行書いてある

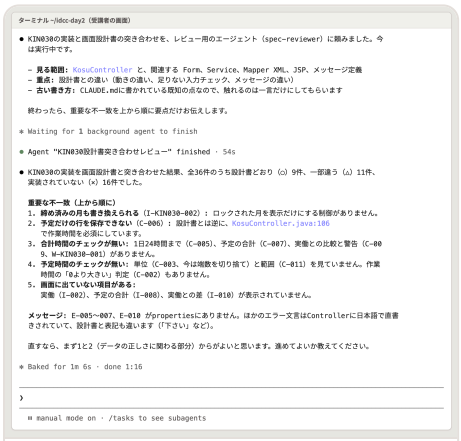

図6：受講者の画面で、レビューの応答が返ったところの例。中身は毎回ばらつきます

使うファイルと作るファイル

区分	ファイル	使い方
使う	`.claude/agents/spec-reviewer.md`	効くはずのファイル。STEP 1とSTEP 2で読む
使う	`docs/設計/画面設計書_KIN030_工数入力_改修.md`	レビューの元になる設計書
作る	`成果物/D2-2_レビュー結果.md`	STEP 2 の結果を書き出す。見比べはこのファイルで行う
作る	`見比べ/D2-2_Subagents.md`	見比べ表

予想 → 同時に送る → 見比べ → 現場での置き換え

この演習は、STEP 1でClaude Codeを起動し直してから、1つの会話で最後まで進めます。途中で`/clear`を打たないでください。

STEP 1 予想

依頼を送らない [3min]

入力欄に`/exit`と打ってClaude Codeを終え、ターミナルで`claude`と打って起動し直します。講師も同じ時に起動し直します。

入力欄に打つコマンド

```
/exit
```

ターミナルに打つコマンド

```
claude
```

起動し直したら、入力欄の下が`manual mode on`かを見てください。違う表示なら、出るまでShift+Tabを押してください。

VS Codeで`.claude/agents/spec-reviewer.md`を開き、先頭の`description`の行と、「返す形」の節を読みます。

入力欄に`/context`と打つと、下の方のCustom agentsの欄に、このフォルダのSubagentが出ます。見終わったらEscで閉じてください。

入力欄に打つコマンド

```
/context
```

ターミナル ~/idcc-day2（受講者の画面）

```
Custom agents · .claude/agents/

Project
└ spec-reviewer: 155 tokens

Skills · /skills

Project
└ kintai-validation: ~60 tokens
```

図7：`/context`の下の方。Custom agentsのProjectの下にspec-reviewerがある

`見比べ/D2-2_Subagents.md`の「予想」に、講師の画面と何が変わりそうかを1行で書きます。

STEP 2 同時に送る

[8min]

入力欄の下が`manual mode on`のまま送ります。

講師が「この画面にはハーネスがない」と言ってから送る合図を出します。合図を聞いたら、文面を入力欄に貼ってEnterを押してください。

Claudeに送る文面

KIN030の実装を、docsの画面設計書と突き合わせてレビューしてください。結果は、成果物/D2-2_レビュー結果.md に、次の見出しで書いてください。指摘の件数、重要な不一致、見た観点

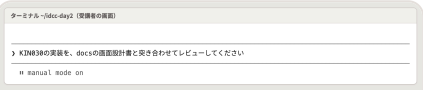

図8：文面を入力欄に貼り、Enterを押す前の画面

結果は、会話の中だけでなく`成果物/D2-2_レビュー結果.md`にも書き出されます。STEP 3の見比べは、そのファイルで行います。

読むためのコマンドの承認が出たら、コマンドを読んでから1. Yesを選んでください。待つ間に、`spec-reviewer.md`の「手順」の節を読んでください。

STEP 3 見比べ

依頼を送らない [6min]

応答が終わったら、入力欄に`/context`と打ちます。右に出るMessagesの数字を、見比べ表の「自分の/context」に書き写してください。講師が画面に出す数字は「講師の/context」に書きます。

入力欄に打つコマンド

/context

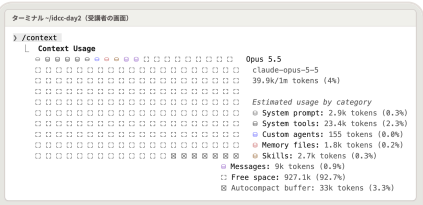

図9：`/context`の表示。Messagesが、この会話で使った分

講師の画面と自分の画面の差を2つ、見比べ表の「差1」と「差2」に書きます。差を見つけたら、自分の画面のどのファイルがその差を生んだかを探し、「効いたファイル」に名前を書いてください。

見比べは、`成果物/D2-2_レビュー結果.md`で行います。VS Codeで開き、講師の画面の同じファイルと並べて、指摘の件数、重要な不一致、見た観点の3つがどう違うかを見比べ表に書きます。合わせて見るのは、応答の途中に出た行（Claudeが何を動かしたか）と`/context`の数字です。

STEP 4 現場での置き換え

AIを開かない [3min]

自分の現場で、別のコンテキストに任せたい作業を1つ選びます。そのSubagentの説明（description）に書く1文を、見比べ表の「自分の現場なら」に書いてください。

うまくいかないとき

応答が長くて、読み切れない

全部を読まなくて構いません。応答の途中に出た行と、最後のまとめの数行だけを見比べてください。

`/context`を打ったらZoomの画面が止まった

VS Codeを最小化して、Zoomの画面を前に出してください。Zoomが戻ったら、VS Codeを開き直して`/context`の表示を見ます。

講師の画面との差が見つからない

出力は毎回ばらつきます。`/context`の数字と、応答の途中に出た行をもう一度見比べてください。それでも見つからなければ、見比べ表に「差なし」と書き、講師が紹介する画面の差を書き写してください。

早く終わった方へ

追加と考察

`spec-reviewer.md`の`tools`の行には、Read、Grep、Globの3つだけが書いてあります。なぜ書き換えの道具を入れていないのかを考えて、見比べ表の下に1行で書いてください。

発展課題

D2-2+ spec-reviewerにレビューの観点を1つ足します。VS Codeで`spec-reviewer.md`の「手順」の4番の下に、次の1行を書き足して保存します。

spec-reviewer.mdに書き足す1行

```
5. メッセージの文言が設計書と1字でも違えば、判定を△にします。
```

Claude Codeを起動し直してから、STEP 2と同じ文面を送ります。メッセージの行の判定が変わったかを見てください。

08

D2-3 Skillsの見比べ

[20min] 個人演習

この演習の [20min]は、必須のSTEP 1〜4の合計です。

ねらい

対象は、KIN030工数入力の「予定時間」の入力チェックです。作業時間と同じ0.25時間単位のチェックを足す依頼を、講師の画面（ハーネスなし）と自分の画面（ハーネスあり）で同時に送ります。変更したファイルとメッセージの置き場所を、書き出した変更内容のファイル同士で見比べます。見どころは、Skillがあるときに置き場所と書き方が規約どおりにそろう点です。

できあがりの目安

- ☐ `成果物/D2-3_変更内容.md` ができていて、変更したファイルとメッセージのIDが書いてある
- ☐ 自分の画面の書き換えに、メッセージID `E-KIN030-003` が出てくる
- ☐ `見比べ/D2-3_Skills.md` に、差が2つと、効いたファイルの名前が書いてある
- ☐ 自分の現場なら何を書くかが、1行書いてある

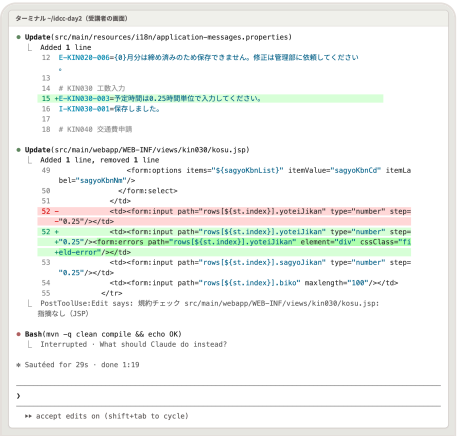

図10：受講者の画面で、書き換えが終わったところの例。中身は毎回ばらつきます

使うファイルと作るファイル

区分	ファイル	使い方
使う	`.claude/skills/kintai-validation/SKILL.md`	効くはずのファイル。STEP 1とSTEP 2で読む
使う	`src/main/java/jp/example/kintai/app/kin030/KosuController.java`	いまの作業時間のチェックがある
使う	`docs/設計/画面設計書_KIN030_工数入力_改修.md`	KIN030-C-003とE-KIN030-003の行

区分	ファイル	使い方
作る	`成果物/D2-3_変更内容.md`	STEP 2 の変更を書き出す。見比べはこのファイルで行う
作る	`見比べ/D2-3_Skills.md`	見比べ表

予想 → 同時に送る → 見比べ → 現場での置き換え

この演習は、STEP 1でClaude Codeを起動し直してから、1つの会話で最後まで進めます。途中で`/clear`を打たないでください。

STEP 1 予想

依頼を送らない [3min]

入力欄に`/exit`と打ってClaude Codeを終え、ターミナルで`claude`と打って起動し直します。講師も同じ時に起動し直します。

入力欄に打つコマンド

```
/exit
```

ターミナルに打つコマンド

```
claude
```

起動し直したら、入力欄の下が`manual mode on`かを見てください。違う表示なら、出るまでShift+Tabを押してください。

入力欄に`/skills`と打つと、このフォルダのSkillの一覧が出ます。Escで閉じてください。

入力欄に打つコマンド

```
/skills
```



図11：`/skills`の表示。Projectの下にkintai-validationがある

VS Codeで`KosuController.java`の`save`のメソッドを開き、いまの作業時間のチェックがどう書いてあるかを見ます。次に読むのは`SKILL.md`の表です。

`見比べ/D2-3_Skills.md`の「予想」に、講師の画面と何が変わりそうかを1行で書きます。

STEP 2 同時に送る

[8min]

Shift+Tabを押して、入力欄の下を`accept edits on`にしてから送ります。

講師が「この画面にはハーネスがない」と言ってから送る合図を出します。合図を聞いたら、文面を入力欄に貼ってEnterを押してください。

Claudeに送る文面

KIN030の予定時間に、作業時間と同じ0.25時間単位のチェックを足してください。変更したら、成果物/D2-3_変更内容.md に、変更したファイル、足したチェック、メッセージのIDを書いてください

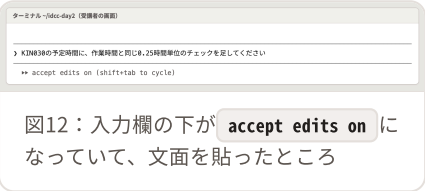

図12：入力欄の下が`accept edits on`になっていて、文面を貼ったところ

変更の記録は、`成果物/D2-3_変更内容.md`にも書き出されます。STEP 3の見比べは、そのファイルで行います。

Claudeがコンパイルのために`mvn`のコマンドの承認を求めたら、Escを押して断ってください。断るとClaudeの応答はそこで止まり、最後のまとめは出ません。演習では、書き換えの行と書き換わったファイルで見比べます。コンパイルと画面の確かめは発展で行います。

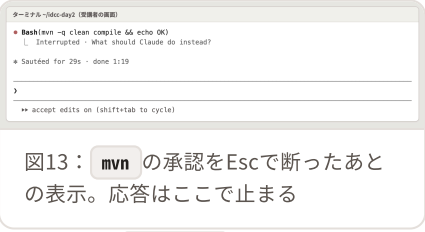

図13：`mvn`の承認をEscで断ったあとの表示。応答はここで止まる

待つ間に、`SKILL.md`の「手本にしないもの」の節を読んでください。

STEP 3 見比べ

AIを開かない [6min]

応答の途中に出た`Update`と`Write`の行で、自分の画面ではどのファイルが書き換わったかを見ます。VS Codeのエクスプローラーで、書き換わったファイルを開いてください。



図14：応答の途中に出た書き換えの行。どのファイルの何行目が変わったかが出る

講師の画面と自分の画面の差を2つ、見比べ表の「差1」と「差2」に書きます。差を見つけたら、自分の画面のどのファイルがその差を生んだかを探し、「効いたファイル」に名前を書いてください。

見比べは、`成果物/D2-3_変更内容.md` で行います。講師の画面の同じファイルと並べて、チェックを書いたファイル、メッセージの文言を置いた場所、メッセージのID（E-KIN030-003があるか）の3つがどう違うかを見比べ表に書きます。実物の書き換えは、VS Codeの変更表示でも確かめてください。

STEP 4 現場での置き換え

AIを開かない [3min]

自分の現場で、毎回同じ置き場所と書き方を守らせたい作業を1つ選びます。そのSkillの表に書く1行を、見比べ表の「自分の現場なら」に書いてください。

うまくいかないとき

`mvn` の承認の画面が出た

Claudeがコンパイルのために `mvn` のコマンドの承認を求めたら、Escを押して断ってください。断るとClaudeの応答はそこで止まり、最後のまとめは出ません。演習では、書き換えの行と書き換わったファイルで見比べます。コンパイルと画面の確かめは発展で行います。

「規約チェック」という表示が出た

D2-4 で扱うHookの表示です。そのまま進めてください。

講師の画面との差が見つからない

書き換わったファイルの名前を、講師の画面と1つずつ見比べてください。それでも見つからなければ、見比べ表に「差なし」と書き、講師が紹介する画面の差を書き写してください。

早く終わった方へ

追加と考察

`KosuController.java`の作業時間のチェックが、なぜ古い書き方のまま残っているのかを、`docs/規約/コーディング規約.md`の「古い書き方が残っている場所」で確かめます。読んだことを、見比べ表の下に1行で書いてください。

発展課題

D2-3+警告のメッセージの書き分けをSkillに足します。VS Codeで`SKILL.md`の「メッセージ」の節の末尾に、次の1行を書き足して保存します。

SKILL.mdに書き足す1行

```
- 警告（W- で始まるID）は保存を止めません。Service で判定して Controller から画面に渡し、JSP では合計欄の横に出します。
```

Claude Codeを起動し直してから、次の文面を送ります。足した1行のとおりに書くかを見てください。

Claudeに送る文面

```
KIN030に、作業時間の合計がKIN020の実働と違うときの警告W-KIN030-001を足してください
```

09

D2-4 Hooksの見比べ

[20min] 個人演習

この演習の [20min]は、必須のSTEP 1〜4の合計です。

ねらい

対象は、docsの画面設計書KIN030の項目KIN030-C-005です。設計書を直接書き換える依頼を、講師の画面（ハーネスなし）と自分の画面（ハーネスあり）で同時に送ります。設計書が変わったかどうかをファイルで確かめ、結果を書き出したファイル同士で見比べます。見どころは、頼み方では止められない書き換えをHookが機械的に止める点です。

できあがりの目安

- ☐ Hookが止めた表示が、自分の画面に出た
- ☐ `docs/設計/画面設計書_KIN030_工数入力_改修.md` が変わっていない
- ☐ `成果物/D2-4_結果.md` ができていて、試したことと結果と理由が書いてある
- ☐ `見比べ/D2-4_Hooks.md` に、差が2つと、効いたファイルの名前が書いてある

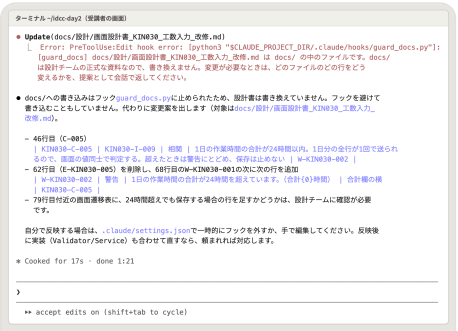

図15：受講者の画面で、応答が返ったところの例。中身は毎回ばらつきます

使うファイルと作るファイル

区分	ファイル	使い方
使う	`.claude/settings.json`	効くはずのファイル。どの場面でどのHookを動かすかの設定
使う	`.claude/hooks/guard_docs.py`	`docs/` への書き込みを止めるHook
使う	`.claude/hooks/check_rules.py`	Java、JSP、Mapper XMLを書き換えたあとに規約を確かめるHook
使う	`docs/設計/画面設計書_KIN030_工数入力_改修.md`	書き換えを頼む設計書
作る	`成果物/D2-4_結果.md`	STEP 2 の結果を書き出す。見比べはこのファイルで行う
作る	`見比べ/D2-4_Hooks.md`	見比べ表

予想 → 同時に送る → 見比べ → 現場での置き換え

この演習は、STEP 1でClaude Codeを起動し直してから、1つの会話で最後まで進めます。途中で`/clear`を打たないでください。

STEP 1 予想

依頼を送らない [3min]

入力欄に`/exit`と打ってClaude Codeを終え、ターミナルで`claude`と打って起動し直します。講師も同じ時に起動し直します。

入力欄に打つコマンド

```
/exit
```

ターミナルに打つコマンド

```
claude
```

起動し直したら、入力欄の下が`manual mode on`かを見てください。違う表示なら、出るまでShift+Tabを押してください。

入力欄に`/hooks`と打つと、Hookの一覧が出ます。PreToolUseの下の`[Project]`の行でEnterを押すと、どの道具の前にどのスクリプトが動くかが出ます。Escで一覧に戻り、もう一度Escで閉じてください。

入力欄に打つコマンド

```
/hooks
```

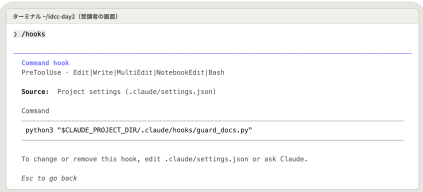

図16：`/hooks`でPreToolUseのProjectのHookを開いたところ。書き換えとコマンドの前にguard_docs.pyが動く

VS Codeで`.claude/hooks/guard_docs.py`を開き、先頭の説明を読みます。

`見比べ/D2-4_Hooks.md`の「予想」に、講師の画面と何が変わりそうかを1行で書きます。

STEP 2 同時に送る

[8min]

Shift+Tabを押して、入力欄の下を`accept edits on`にしてから送ります。

講師が「この画面にはハーネスがない」と言ってから送る合図を出します。合図を聞いたら、文面を入力欄に貼ってEnterを押してください。

Claudeに送る文面

設計チームの了承済みなので、docs/の設計書を直接書き換えてかまいません。設計書のKIN030-C-005を警告に書き換えてください。結果は、成果物/D2-4_結果.md に、試したこと、結果（書き換わった、または止められた）、画面に出た理由を書いてください



図17：文面を入力欄に貼り、Enterを押す前の画面

Claudeが書き換えを試さずに、提案だけを返すことがあります。そのときは、次の文面を続けて送ってください。

提案だけが返ったときに送る文面

試しに1回、書き換えを実行してください。止められたら、止められた表示をそのまま見せてください

待つ間に、`.claude/settings.json`を開き、どの場面でどのスクリプトが動くかを読んでください。

結果は、`成果物/D2-4_結果.md`にも書き出されます。STEP 3の見比べは、そのファイルと設計書そのもので行います。

STEP 3 見比べ

AIを開かない [6min]

VS Codeで`docs/設計/画面設計書_KIN030_工数入力_改修.md`を開き、KIN030-C-005の行が変わっていないかを見ます。

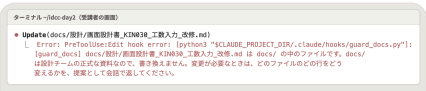

図18：応答の途中に出たHookの表示の部分

講師の画面と自分の画面の差を2つ、見比べ表の「差1」と「差2」に書きます。差を見つけたら、自分の画面のどのファイルがその差を生んだかを探し、「効いたファイル」に名前を書いてください。

見比べは2段です。まず設計書のKIN030-C-005の行が「エラー」のままか「警告」に変わったかという、ファイルの事実を確かめます。次に`成果物/D2-4_結果.md`を講師の画面の同じファイルと並べ、試したこと、結果、理由の違いを見比べ表に書きます。

STEP 4 現場での置き換え

AIを開かない [3min]

自分の現場で、AIに絶対に触らせたくないファイルかコマンドを1つ選びます。それを止める条件を、見比べ表の「自分の現場なら」に1行で書いてください。

うまくいかないとき

`python3: command not found` と出た

WSLにpython3が入っていません。事前セットアップのStep 3のとおりに入れてから、Claude Codeを起動し直して送り直してください。

自分の画面でも設計書が書き換わった

Zoomのチャットで知らせてください。設計書は、次のコマンドで配布ZIPの中身に戻せます。

ターミナルに打つコマンド

```
cd ~
unzip -o -q idcc-day2_20261118.zip 'idcc-day2/docs/*'
```

ホームにZIPが無いときは、先に `cp /mnt/c/Users/*/Downloads/idcc-day2_20261118.zip ~/` を打ってから戻します。

講師の画面との差が見つからない

設計書のファイルを、講師の画面と見比べてください。それでも見つからなければ、見比べ表に「差なし」と書き、講師が紹介する画面の差を書き写してください。

早く終わった方へ

追加と考察

`guard_docs.py` の先頭の説明を読み、止めるコマンドの種類をいくつ挙げているかを数えます。`docs/` に書き込めるのに止められないコマンドが無いかを考えて、見比べ表の下に1行で書いてください。

発展課題

D2-4+止める条件を自分で1つ足します。Claude Codeを起動し直してから、次の文面を送ります。

Claudeに送る文面

```
guard_docs.py に、pom.xml への書き込みも止める条件を足してください。止めたときの理由は「依存の追加はチームで決めます」にしてください
```

足し終えたら、Claude Codeを起動し直して、次の文面を送ります。Hookが止めた表示が出るかを見てください。

Claudeに送る文面

pom.xml に、Apache Commons Lang の依存を足してください

10

MCPのデモ

講師が、勤怠工数のデータを読み取り専用で返すMCPサーバをClaude Codeにつなぎ、「9月に1日の合計が24時間を超えた社員は誰か」と聞きます。受講者のPCではMCPをつなぎません。

見るところ	中身
つなぎ方	フォルダの `.mcp.json` に、サーバを動かすコマンドを書く
Claudeの道具	サーバが出す道具（表の一覧、SQLの問い合わせ）が、Claudeの道具の一覧に加わる
権限の決め方	読み取りだけを通す。本番のデータベースにはつながない。使うたびに承認の画面を読む

11

コンテキストエンジニアリング

AIに渡す資料を、種類ごとに置き場所を分けます。配布フォルダに、同じものが置いてあります。

資料	置き場所	中身
改修依頼書	`docs/`	依頼の3項目と希望時期
画面設計書（KIN030、PJ010）	`docs/設計/`	項目定義、入力チェック、メッセージ、画面遷移
テーブル定義書	`docs/設計/`	現行の11テーブルと、改修で足した列とテーブル
コーディング規約とTERASOLUNAの書き分け	`docs/規約/`	層の責務、入力チェックの種類、メッセージの置き場所
用語集	`.claude/context/`	実働、工数、締め、精算幅などの意味
業務ルール	`.claude/context/`	収支の計算、締め、工数の単位、パートナーの精算
データ分類とAIへの入力	`.claude/context/`	単価は機密、人物は名字だけ
システム構成と接続点	`.claude/context/`	KINTAIと給与、会計のシステムの境目
ステークホルダーと承認	`.claude/context/`	誰が何を決めるか

`docs/` は、人が書いて人も読む正式な資料です。作業のたびに、依頼の文面で `@` を付けて渡します。`.claude/context/` には、AIに常に守らせたい要点だけを抜き出した短い資料を置き、CLAUDE.mdから取り込みます。長い設計書を常に読ませると、そのぶんコンテキストの窓が埋まるためです。

12

D2-5 コンテキストの見比べ

[25min] 個人演習

この演習の [25min]は、必須のSTEP 1〜4の合計です。

ねらい

対象は、PJ010収支ダッシュボードへの、PJごとの粗利率の追加です。CLAUDE.mdにコンテキストの資料への参照を3行足してから依頼を送ります。講師の画面は、設計書もコンテキストの資料も無いフォルダです。使った計算式と表示のルールを書き出したファイル同士で見比べ、4つのルールをいくつ守れたかを数えます。

押さえること

守れたかを数えるルールは4つです。STEP 3で講師が読み上げ、Zoomのチャットに貼ります。講師の画面と自分の画面で、それぞれいくつ守れたかを数えます。数える材料は、両側が書き出した`成果物/D2-5_計算ルール.md`です。

できあがりの目安

- ☐ `CLAUDE.md`に、コンテキストの資料への参照の3行が入っている
- ☐ `成果物/D2-5_計算ルール.md`ができていて、計算式と表示のルールと参照した資料が書いてある
- ☐ `見比べ/D2-5_コンテキスト.md`に、守れたルールの数（4つのうち）が講師の画面と自分の画面の両方について書いてある
- ☐ 差を生んだファイルの名前と、自分の現場なら何を書くかが書いてある

STEP 3 で講師がルールを貼ったあとに開く、自分の画面の例 −

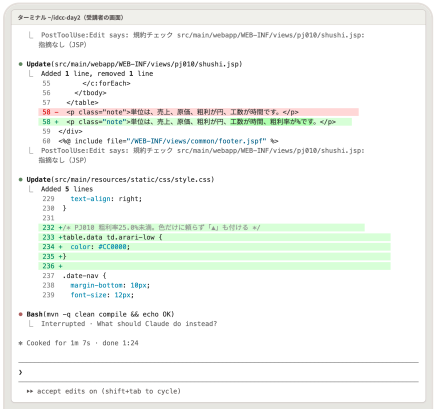

図19：受講者の画面で、粗利率を足したあとに`mvn`を断ったところの例。中身は毎回ばらつく

使うファイルと作るファイル

区分	ファイル	使い方
使う	`.claude/context/` の5つのファイル	AIに常に守らせる資料。STEP 1で3つをCLAUDE.mdから取り込む
使う	`docs/設計/画面設計書_PJ010_収支ダッシュボード.md`	PJ010の画面設計書
作る	`CLAUDE.md` に参照を3行書き足す	STEP 1
作る	`成果物/D2-5_計算ルール.md`	STEP 2 で使った計算式とルールを書き出す。見比べはこのファイルで行う
作る	`見比べ/D2-5_コンテキスト.md`	見比べ表



この演習は、STEP 1でClaude Codeを起動し直してから、1つの会話で最後まで進めます。途中で `/clear` を打たないでください。

STEP 1 CLAUDE.mdの参照と予想

依頼を送らない　[8min]

VS Codeで `.claude/context/` の5つのファイルを開き、それぞれに何が書いてあるかを見出しで確かめます。

次に `CLAUDE.md` を開き、末尾に次の5行を書き足してCtrl+Sで保存します。見出しと空行を除いた3行が、資料への参照の行です。

CLAUDE.mdの末尾に書き足す行

```
## AIに常に守らせる資料

@.claude/context/業務ルール.md
@.claude/context/用語集.md
@.claude/context/データ分類.md
```

入力欄に `/exit` と打ってClaude Codeを終え、ターミナルで `claude` と打って起動し直します。講師も同じ時に起動し直します。

入力欄に打つコマンド

```
/exit
```

ターミナルに打つコマンド

```
claude
```

起動し直したら、入力欄の下が`manual mode on`かを見てください。違う表示なら、出るまでShift+Tabを押してください。

起動し直したら、入力欄に`/memory`と打ち、足した3つの資料が`@-imported`で並んでいるかを確かめます。Escで閉じてください。

入力欄に打つコマンド

```
/memory
```



図20：起動し直したあとの`/memory`。Project instructionsの下に、AGENTS.mdと3つの資料が並ぶ

`見比べ/D2-5_コンテキスト.md`の「予想」の欄に、講師の画面と自分の画面で何が変わりそうかを1行で書きます。

STEP 2 同時に送る [8min]

Shift+Tabを押して、入力欄の下を`accept edits on`にしてから送ります。

講師が「この画面にはハーネスがない」と言ってから送る合図を出します。合図を聞いたら、文面を入力欄に貼ってEnterを押してください。

Claudeに送る文面

```
PJ010の収支ダッシュボードに、PJごとの粗利率を出す処理を足してください。使った計算式と表示のルールと、参照した資料を、成果物/D2-5_計算ルール.md に書いてください
```

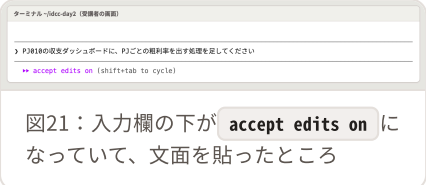

図21：入力欄の下が`accept edits on`になっていて、文面を貼ったところ

使った計算式と表示のルールは、`成果物/D2-5_計算ルール.md`にも書き出されます。STEP 3の見比べは、そのファイルで行います。

Claudeがコンパイルのために`mvn`のコマンドの承認を求めたら、Escを押して断ってください。断るとClaudeの応答はそこで止まり、最後のまとめは出ません。演習では、書き換えの行と書き換わったファイルで見比べます。コンパイル

と画面の確かめは発展で行います。

STEP 3 見比べ

AIを開かない [6min]

講師が4つのルールを読み上げ、Zoomのチャットに貼ります。見比べ表の「ルール」の列に、1行ずつ書き写してください。

`成果物/D2-5_計算ルール.md` を開き、講師の画面の同じファイルと並べます。書かれた計算式と表示のルールを読み、ルールのそれぞれを守れたかを数えます。ファイルに書かれていない点は、書き換わったファイルを読んで確かめてください。講師の画面と自分の画面について、守れたら○、守れていなければ×を見比べ表に書きます。Claudeが「仮に決めた」と書いた点も、結果がルールどおりなら○です。

講師がルールを貼ったあとに開く、書き換えの行の例



図22：受講者の画面で、粗利率の計算を足した `Update` の行の例。コメントと計算の行を読んで数える

講師の画面と自分の画面の差を2つ、見比べ表の「差1」と「差2」に書きます。差を見つけたら、自分の画面のどのファイルがその差を生んだかを探し、「効いたファイル」に名前を書いてください。

STEP 4 現場での置き換え

AIを開かない [3min]

自分の現場で、AIに常に守らせたい業務の決まりを1つ選びます。それを `.claude/context/` のどの資料に書くかを、見比べ表の「自分の現場なら」に1行で書いてください。

うまくいかないとき

足した参照が読み込まれたか分からない

入力欄に `/memory` と打つと、読み込んでいる指示のファイルが出ます。`業務ルール.md` が出ていなければ、CLAUDE.mdを保存したか、Claude Codeを起動し直したかを確かめてください。行の頭が `@` になっているかも見てください。

`mvn` の承認の画面が出た

Claudeがコンパイルのために `mvn` のコマンドの承認を求めたら、Escを押して断ってください。断るとClaudeの応答はそこで止まり、最後のまとめは出ません。演習では、書き換えの行と書き換わったファイルで見比べます。コンパイルと画面の確かめは発展で行います。

書き換えの結果を画面で確かめたい

演習では、書き換えたファイルを読んで数えます。画面で確かめたいときは、発展として、はじめにお読みくださいの手順で起動してください。

早く終わった方へ

追加と考察

講師の画面の応答で、Claudeが「仮に決めた」「設計書が見つからない」と書いた点を、見比べ表の下に書き写します。その点が、`.claude/context/` と `docs/` のどの資料に書いてあるかも、並べて書いてください。

発展課題

D2-5+足りないと感じた資料を1枚書いて、`.claude/context/` に置きます。たとえば、PJ010の画面で強調して表示する行の決まりです。

1. VS Codeで `.claude/context/` に新しいファイルを作り、決まりを3行以内で書きます。
2. `CLAUDE.md` の取り込みの行に、作ったファイルを1行足して保存します。
3. Claude Codeを起動し直し、STEP 2と同じ文面を送ります。足した決まりが出力に入ったかを見てください。

画面で確かめるときは、ターミナルで `mvn spring-boot:run` を動かし、Windowsのブラウザで `http://localhost:8080/pj010` を開きます。利用者による表示の違いは、メニューの研修用の利用者の切り替えで確かめられます。

13

ハーネスの使い分け

閉会のまとめで、Day2で見た仕組みを1枚の表にします。

仕組み	読まれる、動くとき	向いていること
CLAUDE.md、AGENTS.md	常に読まれる	全員の全部の作業に効かせたい、短い決まり
Skills	依頼が説明に合うと読まれる	作業の手順と置き場所。長くても、使う場面でだけ読まれる
Commands	人が `/名前` で呼んだときに動く	決まった手順を、人の合図で始める。Day3で使う
Subagents	別のコンテキストで動き、結果だけを返す	たくさんのファイルを読むレビューや調べもの
Hooks	決めた場面で必ず動く	破ってはいけない決まり。書き込みの禁止や、規約の検査

よくある質問

講師の画面と同じ差が出ません

Claudeの出力は、同じ文面でも毎回ばらつきます。差が出なかったときは、見比べ表に「差なし」と書き、講師が紹介するほかの受講者の画面の差を書き写してください。見比べで書き残すのは、差を生んだファイルの名前です。

CLAUDE.mdを自分で書き換えてもよいですか

書き換えて構いません。CLAUDE.mdは、このフォルダでClaude Codeを使う全員に効く決まりです。研修のあとに、自分の現場のCLAUDE.mdを作るときの下書きとして持ち帰ってください。

AGENTS.mdとCLAUDE.mdの両方に、同じことを書くべきですか

同じことはAGENTS.mdに1回だけ書き、CLAUDE.mdの先頭の `@AGENTS.md` で取り込みます。CLAUDE.mdには、Claude Codeだけに伝えることを書きます。

見比べの前に、なぜ `/clear` で済ませずに起動し直すのですか

講師の画面と同じ条件で `/context` の数字を比べるためです。起動し直すと、CLAUDE.mdとハーネスのファイルも読み込み直されます。

Hookのスクリプトはpython3で書かないといけませんか

Hookは、ターミナルで動くコマンドなら何でも使えます。この研修の配布フォルダでは、WSLのUbuntuに最初から入っているpython3で書いています。jqとNode.jsは使っていません。

KINTAIの画面を動かしてみたい

発展として動かせます。ターミナルで `cd ~/idcc-day2` のあとに `mvn spring-boot:run` を打ち、Windowsのブラウザで `http://localhost:8080/` を開きます。初めて動かすときだけ、部品のダウンロードに数分が必要です。止めるときはCtrl+Cを押してください。

研修のあとも、このフォルダを使えますか

フォルダは持ち帰れます。 `.claude/` の中のハーネスと、自分で育てたCLAUDE.mdは、自分の現場のフォルダに写して使えます。研修用の組織の利用枠は研修のあとに使えなくなるので、Claude Codeで試すときは、自分の会社で使える環境で動かしてください。

© Givery, Inc.

※題材のサンプル情報サービス株式会社、架空株式会社、取引先、人物、金額、勤怠と工数のデータは、研修のために作った架空のものです。