

Day3

AI駆動開発
Claude Code研修

理想的なハーネスと、ハーネスの作り方

株式会社インフォメーション・ディベロプメント様
2026年11月25日（水） オンライン

Givery

本資料の二次転載禁止について

本資料の著作権は株式会社ギブリーに帰属します。受講者の学習を目的として提供するものです。

本資料の全部または一部を、当社の許諾なく複製、転載、配布、公衆送信することを禁じます。
社内での共有をご希望の場合は、事務局を通じてご相談ください。

個人での録画・録音はご遠慮ください。

Copyright © Givery, Inc. All rights reserved

講師紹介



人的資本インテリジェンス部門講師・生成AI
エバンジェリスト

安田 光喜

Mitsuki Yasuda

公立はこだて未来大学 大学院（メディアデザイン領域）を卒業し、街づくり会社での複合商業ビルの立ち上げにおける情報インフラ整備やデザイン全般業務の責任として関与。その後デザイン事務所を立ち上げ、飲食店を中心としたデザイン業務や美術館展示のアートコンテンツ開発などを行う。また、地域ブランディングにも注力し、市主催の企画運営業務を行なった。2018年10月『小中学生向けのプログラミング教育』に注目し、函館市で初めての小中学生向けプログラミングスクール『D-SCHOOL北海道』を立ち上げて運営。その後北海道のドラッグストア『サッポロドラッグストアー』からスクールの買収を受け、教育を専門とするグループ会社『株式会社シーラクンス』にて技術開発責任者として運営。スクール設計・運営、20以上の自治体連携（イベントや出張教室運営）、大人向けプログラミングスクールメンターなどさまざまな『次世代IT教育』に取り組む。
Dify公式資料の日本語翻訳や、SecondMeアンバサダーなど生成AI最新トレンドに注力。

得意領域

・生成AIをはじめとした世界最先端トレンド
・生成AIを用いた新規事業の開発、運営、補助
・アプリケーション開発/自動化プログラム環境構築
・教育（大学講師、研修、スクール運営など）

実績

・生成AI研修
・大手企業、外資、教育機関(小中高大）でのITトレンド講演
・自治体と連携した地域教育実績
・上場企業社員対象のリスキリング支援

メッセージ

みなさんがこれから生成AIをパートナーに新しい時代を生きていけるように、3日間全力でサポートしていきます！なにかご不明点などありましたら、講師陣になんでも気軽にご質問ください。

#ClaudeCode #Codex #Cursor #Antigravity
#全国統一生成AI活用技能試験S1
#生成AIパスポート　#G検定

Copyright © Givery, Inc. All rights reserved

本日のゴール

> **1つのプロジェクトにそろえるハーネスを動かして確かめる**
> **うまくいった手順をSkillにし、別のテーマに使い回す**
>
> Skillの出来は問いません。作る手順を、自分の現場に持ち帰ります。

学びを確かめる成果物（すべてidcc-day3/の中）

ファイル	確かめられること
docs/01-requirements-raw.md	発言者とタイムスタンプが全行にある（D3-1）
.claude/skills/のSKILL.md 2本	自分で作り、1回動かしてある（D3-2とD3-3）
会議室予約の試作品	条件4つと工夫1つが入り、画面がZoomのチャットにある（D3-4）

Copyright © Givery, Inc. All rights reserved

本日の進め方

前半で理想のハーネスを動かし、後半で自分のSkillを作ります

順	コマ	時間	作るもの
1	開始　Day1とDay2のおさらい、配布フォルダの展開	[10min]	
2	座学　とあるプロジェクトの理想的なハーネス設計	[20min]	
3	D3-1 理想的なハーネスの試運転	[20min]	docs/01-requirements-raw.md
4	座学　ハーネスの作り方と使い方	[10min]	
5	D3-2 成功体験のSkill化	[20min]	自作のSkill 1本目
6	休憩	[10min]	
7	D3-3 skill-creatorでのSkill作成	[20min]	自作のSkill 2本目
8	座学　プロジェクト設定とユーザー設定	[10min]	
9	D3-4 別テーマでの成果物づくり	[40min]	会議室予約の試作品
10	閉会　3日間のまとめ、質疑、アンケート	[20min]	

Copyright © Givery, Inc. All rights reserved

Day1のおさらい

AIエージェントは道具を使い、結果を見て次の手を決めます

考える
依頼と資料から次の手を決める

→

道具を使う
ファイル操作とコマンド

→

結果を見る
出力やエラーを読む

Day1でしたこと	覚えておくこと
対話型AIとAIエージェント	対話型AIは答えを返して終わる。AIエージェントは上のループを自分で回す
バイブコーディング	言葉だけで画面を1枚作り、言葉で3回以上直した
注意点	もっともらしい誤りは検算で見つける。承認の画面は読んでから押す
責任の持ち方	機密情報と個人情報は入れない。生成物の責任は人が持つ

▶ 45秒の動画で見る

Copyright © Givery, Inc. All rights reserved

Day2のおさらい

ハーネスの種類ごとに、読まれる時と動く場所が違います

種類	読まれる時と動く場所	見比べ
CLAUDE.md	会話の最初に毎回読まれる。プロジェクトの規約を書く	D2-1
Skills	依頼が説明文に合うと本文が読まれる。手順と観点を書く	D2-3
Subagents	別のコンテキストで動き、結果だけを親の会話に返す	D2-2
Hooks	決まった瞬間に必ず動く。書き込みやコマンドを止められる	D2-4
MCP	外の道具やデータベースをつなぐ	講師のデモ
コンテキスト	docs/は作業ごとに@で渡し、.claude/context/はCLAUDE.mdから読ませる	D2-5

今日はこの6つを、1つのプロジェクトの10工程の中で組み合わせて使います。
MCPは研修版の配布フォルダから外してあります。

Copyright © Givery, Inc. All rights reserved

Day3の配布フォルダ

今日は理想的なハーネスが最初から入ったフォルダから始めます

idcc-day3/

├ はじめにお読みください.md	架空の題材であることと、展開と起動の手順
├ CLAUDE.md	全工程の規約。応答を10行以内にする約束と保存先
├ ハーネスの地図.md	D3-1で、予想と結果の印を付ける表
├ .claude/settings.json	権限とHooksの登録。モデルを指定する行は無い
├ .claude/agents/	Subagent 10体
├ .claude/commands/	/extract-requirementsほか、工程をつなぐ3本
├ .claude/skills/ hooks/ templates/	Skill 4本、python3のHook、雛形3本
├ transcripts/	在庫管理のヒアリングの文字起こし（D3-1）
├ data/	3拠点の在庫のCSVと列の説明（D3-2）
├ 別テーマ/	会議室予約のヒアリングメモとサンプルデータ（D3-4）
├ docs/と 成果物/	D3-1からD3-4で作ったものの置き場
└ _予備/skill-creator/	公式から取り込めないときだけ使う写し（D3-3）

架空株式会社（フィクション）の題材です。会社名、人物、数字はすべて架空です。

Copyright © Givery, Inc. All rights reserved

実演

配布フォルダの展開と起動

ダウンロードしたZIPを**ホームに展開**します

Ubuntu22.04のターミナル（VS Codeの左下がWSL: Ubuntu-22.04）

```
echo $WSL_DISTRO_NAME # Ubuntu-22.04 と返ることを確かめる
cd ~
cp /mnt/c/Users/*/Downloads/idcc-day3_20261125.zip ~/
unzip -o idcc-day3_20261125.zip
cd ~/idcc-day3
code .
```

VS Codeが開いたら、統合ターミナルでclaudeと打って起動します

- ☐ VS Codeで~/idcc-day3が開き、CLAUDE.mdと.claude/が見える
- ☐ 信頼の画面は下矢印で Yes, I trust this folderを選ぶ
- ☐ Shift+Tabで accept edits onにしてから始める

Day2のフォルダで動いているclaudeは閉じます

Day2のターミナルで/exitを打ってから、Day3のフォルダで起動し直してください。

Copyright © Givery, Inc. All rights reserved

題材 架空株式会社の在庫管理システム刷新

Excelの在庫台帳3本を1つのシステムに置き換える案件です

項目	内容
会社	架空株式会社（フィクション）。電子部品の専門商社、東京3拠点、社員120名
今の運用	Excelの在庫台帳3本（本社の発注用、江東と大田の入出庫用）。品目は約12,000SKU
困りごと	台帳の突き合わせに毎朝時間がかかる。在庫ありと答えた品が欠品していたことがある
欲しいもの	バーコードでの入出庫、スマホでの在庫照会、販売管理とのCSV連携
制約	予算は年額1,500万円以内。リースPCの満了（来年9月）が実質の期限
配る資料	ヒアリングの文字起こし（transcripts/）と、3拠点の在庫のCSV（data/）

顧客ヒアリングから保守運用までの10工程に、ハーネスを割り当ててあります。

Copyright © Givery, Inc. All rights reserved

ハーネス一式の中身

1つのプロジェクトに6種類のハーネスをそろえてあります

ハーネス	中身	使う工程
CLAUDE.md	要求工学、文書、コーディング、テスト、レビューの規約	全工程
Subagents 10体	req-extractor、req-reviewerほか、工程ごとの専門家	工程ごと
Commands	/extract-requirementsほか、工程をつなぐ3本	01、07、08
Skills 4本	rdd-template、security-review、simplify、postmortem	03、07、08、10
Hooks	危険なコマンドと機密値の遮断、根拠の数え、試作品の検査ほか	全工程と3工程
templates	user-story、adr、error-codeの書式	04、05、06
MCP	研修版では外し、Day2の講師のデモで扱った	なし

工程の番号は次のページの地図と同じです。

▶ 46秒の動画で見る

Copyright © Givery, Inc. All rights reserved

10工程とハーネスの地図

▶ 93秒の動画で見る

どの工程でどのハーネスが働くかを重ねた図です

CLAUDE.md　全工程で毎回読まれる規約（要求工学、文書、試作品のコーディング、確かめ、レビュー）

工程	Subagent	Command	Skillと雛形	Hook	演習
01 ヒアリング解析	req-extractor → req-reviewer	/extract-requirements		根拠の数え	D3-1 D3-4
02 要求の優先度付け	prioritizer				
03 要件定義書作成	req-reviewer（点検）		rdd-template	レビューの関所	
04 仕様定義	spec-writer		user-story（雛形）		
05 基本設計	architect		adr（雛形）		
06 詳細設計	detail-designer		error-code（雛形）		
07 実装	code-reviewer（1回）	試作品づくりのコマンド	simplify	試作品の検査	D3-4
08 テストとレビュー	code-reviewer、qa-engineer	並列レビュー（発展）	security-review		
09 CI/CDとデプロイ	devops				
10 保守運用	incident-responder		postmortem		

Hooks　全工程で、危険なコマンドの遮断と機密値の混入検知を受け持つ。settings.jsonの権限で入力の資料も守る

Copyright © Givery, Inc. All rights reserved

工程01で動く順番

▶ 96秒の動画で見る

1つのコマンドが2体のSubagentを順に呼びます

順	動くもの	すること	残るもの
1	受講者	/extract-requirementsに文字起こしのファイルを渡す	送るのは1行だけ
2	req-extractor	別のコンテキストで文字起こしを読み、要求を表にする	01-requirements-raw.md
3	Hook	書いた直後に、全行に根拠があるかを数える	根拠の無い行の数
4	req-reviewer	別のコンテキストで要求一覧を点検する	01-requirements-review.md
5	Command	[WARN]か[ERROR]があれば次へ進まず、人を待つ	人が[確認済]に書き換える

CLAUDE.md 全部の要求に発言者と根拠と確信度を付ける規約を、毎回読む

Hooks どの段でも、危険なコマンドと機密値の書き込みを止める

Copyright © Givery, Inc. All rights reserved

解説

生成とレビューを分ける理由

書いた本人とは別のコンテキストで点検させます

このフォルダの書く役と点検する役

成果物	書く役	点検する役
要求一覧	req-extractor	req-reviewer
要件定義書	本体の会話と rdd-template	req-reviewer
試作品	本体の会話	code-reviewer

Anthropicのベストプラクティス

新しいコンテキストで見ると、
Claudeは書いたばかりのコードに
引っ張られずにレビューできる

レビュー役が見るのは、結果と基準だけです

原文　“A fresh context improves code review since Claude won't be biased toward code it just wrote.”

配布フォルダのCLAUDE.mdにも、
生成した本人によるセルフレビューを禁じる行があります。

出典　Anthropic「Best practices for Claude Code」Claude Code Docs（2026年10月22日閲覧）
https://code.claude.com/docs/en/best-practices

▶ 57秒の動画で見る

Copyright © Givery, Inc. All rights reserved

Commands、Skills、templatesの役目

人が呼ぶ手順はCommand、条件で読まれる手順はSkillに置きます

種類	置き場	呼ばれ方	このフォルダの例
Commands	.claude/commands/	人が/名前で呼び、 工程の手順をつなぐ	/extract-requirementsほか
Skills	.claude/skills/	依頼が説明文に合うと読まれ、 /名前でも呼べる	rdd-templateほか
templates	.claude/templates/	SubagentやSkillが 書式として読む	user-story、adr、error-code

> Custom commands have been merged into skills.
>
> 公式ドキュメントでは、どちらも/名前 を作る同じ仕組みとして扱います。

出典　Anthropic「Extend Claude with skills」Claude Code Docs（2026年10月22日閲覧）
https://code.claude.com/docs/en/skills

▶ 63秒の動画で見る

Copyright © Givery, Inc. All rights reserved

解説

Hookをガードレールにする理由

必ず止めたい操作はCLAUDE.mdでなくHookに書きます

CLAUDE.mdは頼みごと

Claudeは中身を文脈として読みます。
守らせる仕組みは持っていません。

Hookは必ず動く処理

AIが実行を選ぶかどうかに関係なく、
決めた処理が毎回動きます。

このフォルダのHook	動く瞬間	止めたとき
危険なコマンドの遮断	コマンドを実行する前（PreToolUse）	理由を表示する
機密値の混入検知	ファイルを書く前とコマンドの前（PreToolUse）	書き込みを止める
レビューの関所	次の工程のコマンドを送ったとき（UserPromptSubmit）	指摘が残れば止める
根拠の数え、試作品の検査	ファイルを書いたあと（PostToolUse）	指摘をClaudeに返す

原文　“Claude treats them as context, not enforced configuration.”
原文　“certain actions always happen rather than relying on the LLM to choose to run them”

出典　Anthropic Claude Code Docsのmemoryとhooks-guideのページ（2026年10月22日閲覧）
https://code.claude.com/docs/en/memory　https://code.claude.com/docs/en/hooks-guide

▶ 59秒の動画で見る

Copyright © Givery, Inc. All rights reserved

研修版で直したところ

▶ 98秒の動画で見る

研修の環境で動くように7か所を直してあります

元のハーネス（Giveryの商談用デモ）	研修版
型検査とlintをNext.jsのコマンドで呼ぶHook	題材をHTMLとJavaScriptの試作品にし、python3の検査に替えた
Gitのブランチ、コミット、PRを使う手順	Gitの手順を外し、成果物のフォルダに保存する手順にした
GitHub、Datadog、Slackのトークンが要るMCP	配布フォルダから外した。MCPはDay2の講師のデモで扱った
settings.jsonとSubagent 10体がOpusを固定	どちらも外し、受講者の既定のモデルで動かす
READMEに実物の無いコマンドの記載	研修で使う工程のコマンドだけを置き、READMEをそろえた
Hookのスクリプトがjqを使う	python3の標準ライブラリだけで書き直した
毎朝の定期実行で監視のレポートを作る	監視のサービスとMCPが要るので外した

SubagentとCommandとSkillの組み合わせ方は、元のハーネスのままです。

Copyright © Givery, Inc. All rights reserved

▶ 94秒の動画で見る

Subagentの並列は発展だけにし、モデルは固定しません

決まり	中身
Subagentの並列は発展だけ	必須の演習で同時に動くSubagentは1体まで。 3体の並列レビューはD3-1の発展で1回だけ
モデルを固定しない	settings.jsonとSubagentの定義にモデルの行を書かない。 受講者の既定のモデルで動かす
応答は要点だけ、10行以内	CLAUDE.mdに書いてある。長い表やファイルの中身は、 頼んだときだけ出させる

Subagentの利用量（公式ドキュメント）

It also sends its own requests, which count toward the same usage limits as your main conversation.

Subagentのやり取りも、本体の会話と同じ利用量の上限に数えられます。

出典　Anthropic「Create custom subagents」Claude Code Docs（2026年10月22日閲覧）https://code.claude.com/docs/en/sub-agents

Copyright © Givery, Inc. All rights reserved

D3-1

理想的なハーネスの試運転

[20min]

作るもの

docs/01-requirements-raw.mdとレビュー、ハーネスの地図.mdの印

Copyright © Givery, Inc. All rights reserved

ワーク

演習D3-1のねらいと目安

ねらい

対象は、ヒアリングの文字起こしから要求一覧を作る工程01です。ハーネスを全部そろえたフォルダで工程01を動かし、どのハーネスがどの順に働くかを自分の画面で見ます。

できあがりの目安

- [ ] ハーネスの地図.mdの予想と結果の列に〇が付いている
- [ ] docs/01-requirements-raw.mdの全行に、発言者とタイムスタンプがある
- [ ] 危険なコマンドをHookが止めた表示が、自分の画面に出た

使うファイル・作るファイル

区分	ファイル
追記	ハーネスの地図.md（予想と結果の列）
使う	transcripts/kakuu-2026-04-15.txt（ヒアリングの文字起こし）
作る	docs/01-requirements-raw.mdとdocs/01-requirements-review.md

Copyright © Givery, Inc. All rights reserved

ワーク

D3-1の流れ

STEP1の5分はAIを開かずにフォルダを読みます

■ 自分の手で　■ AIと一緒に　□ 座学・理解度チェック　　演習の枠 [20min]

STEP1 [5min]	STEP2 [10min]	STEP3 [5min]

■ **STEP1　ハーネスの地図**	ハーネスの地図.mdと.claude/の中を読み、 D3-1で動くと思うものの予想に〇を付けます
■ **STEP2　要求の抽出**	/extract-requirementsに文字起こしを渡し、 2体のSubagentが順に動くのを見ます
■ **STEP3　Hookの確かめ**	ガイドの文面で危険なコマンドを頼み、 止めた表示を見て、結果に〇を付けます

Copyright © Givery, Inc. All rights reserved

解説

D3-1で見るところ

画面に出るSubagentの名前とHookの表示を追います

いつ	画面に出ること	確かめること
STEP2	req-extractorの名前と作業中の表示	Subagentが1体ずつ動いている
STEP2	要求一覧を書いた直後のHookの検査	根拠の無い行が0件
STEP2	req-reviewerの[ERROR] [WARN] [INFO]の件数	重い指摘の上位3件を読む
STEP2	10行以内の報告と、次にすること	指摘が残れば次へ進んでいない
STEP3	Hookが止めた理由の表示	コマンドが実行されていない

Subagentは1体ずつ順に動きます

必須の演習で、2体以上が同時に動くことはありません。
同時に動かすのは、発展の並列レビューだけです。

Copyright © Givery, Inc. All rights reserved

発展

D3-1早く終わった方へ

追加と考察

req-reviewerの指摘を1件選び、文字起こしの該当する行に戻って、指摘が正しいかを自分で確かめます。

発展課題

D3-1+　並列レビューのコマンドで、3つのレビューを同時に走らせます。
3体が同時に利用量を使うので、1回だけにします。

時間が余った方だけ取り組みます。答え合わせは講師が口頭で行います。

Copyright © Givery, Inc. All rights reserved

ハーネスの作り方4つ

今日はこのうち2つを自分の手で試します

作り方	中身	今日
公式から持ってくる	AnthropicのGitHubと公式マーケットプレイスの雛形やプラグインを入れる	座学とD3-3
チームで使い回す	プロジェクトに置いた.claude/を、同じフォルダを開く全員で使う	D3-1とD3-4
成功体験を落とす	うまくいった対話の手順を、そのままSkillにする	D3-2
AIと相談して作る	skill-creatorに、何を自動化したいかを話しながら作る	D3-3

上の2つは、できあがったハーネスを手に入れる手です。
下の2つは自分の手順からハーネスを作る手で、今日の後半で両方を試します。

Copyright © Givery, Inc. All rights reserved

公式テンプレートと公認ハーネスの入手先

雛形はAnthropicのGitHubと公式マーケットプレイスにあります

公式テンプレート

anthropics/skills

AnthropicのSkillの実例と雛形。skill-creatorもここにある。
/plugin marketplace add anthropics/skillsで取り込める

https://github.com/anthropics/skills

公認ハーネス

anthropics/claude-plugins-official

公式マーケットプレイス。Anthropicが管理するプラグインと、
パートナーのプラグインが並ぶ。初回の起動で登録される

https://github.com/anthropics/claude-plugins-official

公認ハーネス

Claude Marketplace

プラグインをWebで探せる。導入数と、
Anthropicが確かめた印（Anthropic verified）が出る

https://claude.com/marketplace/plugins

出典　Anthropic「Anthropic's marketplaces」Claude Code Docs（2026年10月22日閲覧）
https://code.claude.com/docs/en/plugins/anthropic-marketplaces　https://github.com/anthropics/skills

Copyright © Givery, Inc. All rights reserved

プラグインの入れ方と使える範囲

▶ 95秒の動画で見る

入れるときに自分だけかチーム全員かを選びます

1. /pluginを打ち、Discoverのタブで探す
2. 詳細を読み、公式のものはトークンの見積もりも見る
3. 範囲を選んで入れる

```
/plugin install skill-creator@claude-plugins-official
```

範囲	使える人	記録されるファイル
ユーザー	このPCの全部のプロジェクトの自分	~/.claude/settings.json
プロジェクト	このプロジェクトで作業する全員	.claude/settings.json
ローカル	このプロジェクトの自分だけ	.claude/settings.local.json

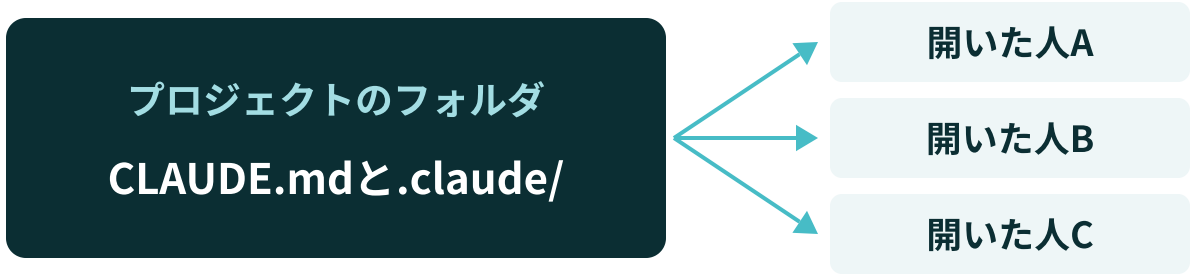
同じプラグインを複数の範囲で入れたときは、ローカルとプロジェクトとユーザーの順に優先されます。

出典　Anthropic「Install plugins」Claude Code Docs（2026年10月22日閲覧）https://code.claude.com/docs/en/discover-plugins

Copyright © Givery, Inc. All rights reserved

チームで作ったハーネスの使い回し

プロジェクトに置いた.claude/は開いた全員に働きます



開いた人ごとにハーネスを設定し直す手間がなくなり、
全員が同じ規約と手順で作業します。

使い回し方	中身
プロジェクトに置く	CLAUDE.mdと.claude/をプロジェクトに入れ、チームのリポジトリで共有する
プラグインにまとめる	複数のプロジェクトで使うSkill、Subagent、Hookを1つにまとめて配る
良かったSkillを集める	チームの共有の場所に置き、使うプロジェクトの.claude/skills/に写す

出典　Anthropic「Settings files and precedence」「Plugins overview」Claude Code Docs（2026年10月22日閲覧）
https://code.claude.com/docs/en/settings　https://code.claude.com/docs/en/plugins

Copyright © Givery, Inc. All rights reserved

補足

公式の雛形を社内で使うときの確かめ

公式の雛形も自分の環境で試してから使います

確かめること	中身
デモと学習のための公開	anthropics/skillsは、デモと学習のための公開だと書き、 大事な作業に使う前に自分の環境で試すよう求めている
Skillごとのライセンス	多くはApache 2.0。docx、pdf、pptx、xlsxのSkillは ソースを公開しているだけで、オープンソースではない
プラグインの中で動くもの	プラグインにはHookやMCPサーバーが入ることがある。 入れる前に中身と、動くコマンドを読む
社内の決まり	外から持ち込むソフトの審査と、組織の管理者が プラグインやSkillを制限していないかを確かめる

出典　anthropics/skillsのREADME（2026年10月22日閲覧）https://github.com/anthropics/skills
Anthropic「Plugins overview」Claude Code Docs https://code.claude.com/docs/en/plugins

▶ 95秒の動画で見る

Copyright © Givery, Inc. All rights reserved

D3-2

成功体験のSkill化

[20min]

作るもの

.claude/skills/<自分で決めた名前>/SKILL.md

Copyright © Givery, Inc. All rights reserved

解説

Skillの読み込まれ方

説明文は毎回読まれ、本文は使うときだけ読まれます

起動したとき

nameとdescriptionだけが入る

rdd-template	要件定義書の章立てを当てる
security-review	差分をOWASPの観点で見る
postmortem	障害の振り返りを書く

依頼 →

依頼が説明文に合ったとき

選ばれたSkillの本文だけを読み込む

postmortem/SKILL.mdの本文

概要とタイムラインと根本原因と
再発防止の4つの章を立てる

「気をつける」は再発防止に数えない

自分から使うときは、/名前の形で呼びます。
本文は呼ばれるまで読まれないので、長い手順を書いても
毎回の会話は重くなりません。

▶ 62秒の動画で見る

出典　Anthropic「Extend Claude with skills」Claude Code Docs（2026年10月22日閲覧）https://code.claude.com/docs/en/skills

Copyright © Givery, Inc. All rights reserved

SKILL.mdの形

▶ 93秒の動画で見る

いつ使うかの説明と手順の2つでできています

.claude/skills/weekly-report/SKILL.md（別の題材の例）

```
---
name: weekly-report
description: 作業ログのCSVから週報の表を
作るときに使う。担当者ごとの合計と
先週との差を出す
---
# 週報の表を作る
1. CSVの日付、担当者、作業時間の列を見る
2. 担当者ごとに今週と先週の合計を出す
3. 担当者、今週、先週、差の4列の表にする
```

name

/weekly-reportで呼べる名前。
フォルダの名前とそろえる

description

Claudeが選ぶときに読む。
何をするかと、いつ使うかを書く

本文

使うときだけ読まれる手順。
番号を付けて短く書く

Copyright © Givery, Inc. All rights reserved

ワーク

演習D3-2のねらいと目安

ねらい

対象は、在庫CSVから発注候補表を作る作業です。うまくいった対話の手順を、そのままSkillにします。
/clearのあとでも、一言で同じ形の表が出るかを、2つの表を並べて確かめます。

できあがりの目安

- [ ] .claude/skills/にSKILL.mdがあり、手順が3つ書いてある
- [ ] SKILL.mdを自分で読み、1か所直してある
- [ ] /clearのあとの一言で、同じ列の表が出る

使うファイル・作るファイル

区分	ファイル
使う	data/在庫_202609.csvとdata/列の説明.md
作る	成果物/D3-2/の発注候補表（STEP1でClaudeと3往復して作る）
作る	.claude/skills/<自分で決めた名前>/SKILL.md

Copyright © Givery, Inc. All rights reserved

ワーク

D3-2の流れ

Skillにする前にClaudeと3往復で表を仕上げます

■ 自分の手で　■ AIと一緒に　□ 座学・理解度チェック　　演習の枠 [20min]

STEP1 [9min]	STEP2 [5min]	STEP3 [6min]

■ **STEP1　3往復で仕上げる**	在庫CSVから安全在庫を下回る品目を抜き、 拠点ごとの発注候補表にします
■ **STEP2　Skillにする**	今のやり方をSkillにするよう頼み、SKILL.mdを読んで1か所直します
■ **STEP3　/clearのあとに試す**	/clearのあと一言だけ頼み、同じ列の表が出るかを見ます

Copyright © Givery, Inc. All rights reserved

解説

D3-2で見る観点

▶ 92秒の動画で見る

説明文にいつ使うかが書いてあるかを見ます

SKILL.mdで見る3つ

- [ ] descriptionに、何をするかと、いつ使うかが書いてある
- [ ] 手順が3つあり、表の列の名前と順まで書いてある
- [ ] 3往復のうちに直したことが、手順に入っている

よくある落とし穴

起きること	原因	直し方
一言で頼んでも選ばれない	descriptionが短すぎる	いつ使うかを足す
列の名前や順がばらつく	SKILL.mdに列の名前がない	列の名前と順を手順に書く
3往復目の直しが消えた	最初の依頼だけを手順にした	直した点を手順に足す

Copyright © Givery, Inc. All rights reserved

発展

D3-2早く終わった方へ

追加と考察

/clearのあと、言い方を変えた一言でも同じSkillが選ばれるかを試します。
選ばれなければ、descriptionを直します。

発展課題

D3-2+　SKILL.mdに確かめの手順を足します。
たとえば、抜き出した品目の件数を数えて、表の下に書く手順です。

時間が余った方だけ取り組みます。答え合わせは講師が口頭で行います。

Copyright © Givery, Inc. All rights reserved

休憩

[10min]

再開は講師の合図でお知らせします

Copyright © Givery, Inc. All rights reserved

D3-3

skill-creatorでのSkill作成

[20min]

作るもの

.claude/skills/<自分で決めた名前>/SKILL.mdと、1回動かした結果

Copyright © Givery, Inc. All rights reserved

skill-creatorの役目

▶ 92秒の動画で見る

Skillの作り方そのものをClaudeに渡す公式のSkillです

1. 何をさせたいか聞き取って決める
2. SKILL.mdの下書きを書く
3. 試しの依頼を送って動かす
4. 結果を見てSKILL.mdを直す

skill-creatorが最初に確かめること

聞かれること	答え方
何をできるようにするか	作業の中身を1文で答える
いつ使うか	どんな依頼や場面で選ばれてほしいかを答える
何を出すか	表の列やファイルの形を答える
確かめのテストを作るか	今日は作らない、と答える。評価の繰り返しは発展に回す

取り込む先は2つあります。GitHubのanthropics/skillsと、公式マーケットプレイスのskill-creatorプラグインです。

出典　anthropics/skillsのskills/skill-creator/SKILL.md（2026年10月22日閲覧）https://github.com/anthropics/skills

Copyright © Givery, Inc. All rights reserved

作りたいSkillの3行

手で書くのは3行で、残りはskill-creatorと決めます

書く3行と、別の題材での例

1	**何を渡すか**	問い合わせのCSVのファイル
2	**何を作るか**	分類ごとの件数を、多い順に並べた表
3	**どこへ保存するか**	成果物/D3-3/問い合わせ_件数.md

3行を渡すと、skill-creatorが足りないところを質問します

答えると、SKILL.mdの下書きができます。
迷う質問には、今日は決めないと答えて先へ進んでかまいません。

Copyright © Givery, Inc. All rights reserved

ワーク

演習D3-3のねらいと目安

ねらい

対象は、自分の業務に近い題材で作るSkillです。skill-creatorと相談しながら1から作ります。
手順を持ち帰るのが目的です。仕上げに、D3-2のSKILL.mdとdescriptionを見比べます。

できあがりの目安

- [ ] 自分で作ったSKILL.mdが.claude/skills/にある
- [ ] descriptionに、いつ使うかが書いてある
- [ ] 作ったSkillを1回動かした結果が、ファイルに保存されている

使うファイル・作るファイル

区分	ファイル
使う	skill-creator（公式。取り込めないときは_予備/skill-creator/）
作る	.claude/skills/<自分で決めた名前>/SKILL.md
作る	成果物/D3-3/の、1回動かした結果

Copyright © Givery, Inc. All rights reserved

ワーク

D3-3の流れ

STEP2では聞かれたことに答えながら作ります

■ 自分の手で　■ AIと一緒に　□ 座学・理解度チェック　　演習の枠 [20min]

STEP1 [4min]	STEP2 [10min]	STEP3 [6min]

■	**STEP1　skill-creatorの取り込み**	公式のskill-creatorをプロジェクトに取り込みます。 取り込めないときは_予備/の写しを使います
■	**STEP2　3行を書いて相談**	作りたいSkillを3行で書き、 skill-creatorの質問に答えながら作ります
■	**STEP3　1回動かす**	作ったSkillを1回動かし、結果をファイルに保存します

テストの依頼を何本も流す評価の繰り返しは、利用量を使うので発展に回します。

Copyright © Givery, Inc. All rights reserved

発展

D3-3早く終わった方へ

追加と考察

作ったSkillのdescriptionを読み、別の言い方の依頼でも選ばれるかを1回試します。

発展課題

D3-3+　作ったSkillのフォルダを~/.claude/skills/に写します。
別のフォルダで起動したclaudeでも動くかを確かめます。

時間が余った方だけ取り組みます。答え合わせは講師が口頭で行います。

Copyright © Givery, Inc. All rights reserved

設定の置き場と使える範囲

同じ名前のファイルでも置き場で使える範囲が変わります

置き場	使える範囲	置くもの
.claude/settings.jsonと CLAUDE.md	そのフォルダを開いた全員	チームで守る規約、権限、Hooks
.claude/settings.local.jsonと CLAUDE.local.md	そのフォルダを開いた自分だけ	自分の環境に依存する設定
~/.claude/settings.jsonと ~/.claude/CLAUDE.md	自分が開く全部のフォルダ	口調、自分の好み、 どの案件でも使うSkill

Skillも同じ分け方で、.claude/skills/と~/.claude/skills/に置きます。
Subagentの置き場は.claude/agents/と~/.claude/agents/です。

> **組織の管理者が配る設定（managed settings）**
>
> どのファイルよりも優先されます。どの設定が読まれたかは/statusで見られます。

出典　Anthropic「Settings files and precedence」Claude Code Docs（2026年10月22日閲覧）https://code.claude.com/docs/en/settings

Copyright © Givery, Inc. All rights reserved

重なったときの優先順位

重なり方はファイルの種類ごとに違います

重なるもの	どれが使われるか
settings.jsonの同じ項目	組織の管理>コマンドライン>ローカル>プロジェクト>ユーザー。 許可のリスト（permissions.allow）は足し合わされる
CLAUDE.md	上書きせず全部をつなげて読む。ユーザーのものが先に入り、 プロジェクトとCLAUDE.local.mdが後に読まれる
同じ名前のSkill	組織>ユーザー>プロジェクト。 ~/.claude/skills/のほうが動く
同じ名前のSubagent	組織>コマンドライン>プロジェクト>ユーザー>プラグイン。 .claude/agents/のほうが動く

SkillとSubagentでは、優先される側が逆になります。
同じ名前を両方に置かないでください。

出典　Claude Code Docsのsettings、memory、skills、sub-agentsのページ（2026年10月22日閲覧）
https://code.claude.com/docs/en/settings　/memory　/skills　/sub-agents

▶ 56秒の動画で見る

Copyright © Givery, Inc. All rights reserved

プロジェクトとユーザーに置くもの

チームで守ることはプロジェクトに、自分の好みはユーザーに置きます

プロジェクト

CLAUDE.mdと.claude/

要求工学とレビューの規約

権限とHooks

Subagent 10体とSkill 4本

D3-2で作った、
この案件の手順のSkill

ユーザー

~/.claude/

口調と応答の長さ

どの案件でも使う自分のSkill

D3-3で作ったSkillを、
全部のフォルダで使うとき

ローカル

settings.local.jsonと
CLAUDE.local.md

自分のPCだけのパス

試している途中の設定

D3-4の発展では、~/.claude/CLAUDE.mdに自分の好みを書き、成果物が変わるかを見ます。

Copyright © Givery, Inc. All rights reserved

使われている設定の確かめ方

迷ったら読み込まれたファイルを画面で確かめます

確かめたいこと	打つもの	見るところ
読み込まれた設定ファイル	/status	Setting sourcesの行
読み込まれたCLAUDE.md	/memory	ユーザーとプロジェクトのファイルの一覧
使えるSkill	/skills	名前、説明、置き場
Hookの設定	/hooks	どのイベントで何が動くか
入れたプラグインと範囲	/plugin	Installedのタブ

/statusの画面は、どのファイルを読んだかを示します。どの項目がどのファイルから来たかまでは出ません。

出典　Anthropic「Commands」「Settings files and precedence」Claude Code Docs（2026年10月22日閲覧）
https://code.claude.com/docs/en/commands　https://code.claude.com/docs/en/settings

Copyright © Givery, Inc. All rights reserved

D3-4

別テーマでの成果物づくり

[40min]

作るもの

会議室予約の試作品と、Zoomのチャットに貼る画面1枚

Copyright © Givery, Inc. All rights reserved

題材 会議室予約

▶ 102秒の動画で見る

在庫管理とは別の架空の会社で同じハーネスを使います

項目	内容
題材	サンプル電機株式会社（架空）の会議室予約
配る資料	別テーマ/のヒアリングメモ_会議室予約.mdと、 会議室.csv、予約_202612.csv、データの説明.md
作るもの	成果物/D3-4/index.html。HTMLとJavaScriptの1ファイルの試作品
条件4つ	予約の一覧、会議室と日付での絞り込み、重複予約のエラー表示、予約の取り消し
自分の工夫	ヒアリングメモを読み、足すものを自分で1つ決める
使うハーネス	D3-1と同じCLAUDE.md、要求抽出と試作品づくりのコマンド、Subagent、Hook

在庫管理のために組んだハーネスを、そのまま別の案件に使い回します。人によって成果物が違ってかまいません。

Copyright © Givery, Inc. All rights reserved

ワーク

演習D3-4のねらいと目安

ねらい

対象は、会議室予約の要求一覧から試作品までの一式です。配られたハーネスを使い回して作ります。
条件4つを満たしたうえで、自分の工夫を1つ入れます。

できあがりの目安

- [ ] 予約の一覧と、会議室と日付での絞り込みが動く
- [ ] 重複予約でエラーが出て、予約を取り消せる
- [ ] 自分の工夫が1つ入っている
- [ ] 画面1枚と工夫を書いた1行が、Zoomのチャットにある

使うファイル・作るファイル

区分	ファイル
使う	別テーマ/のヒアリングメモ、会議室.csv、予約_202612.csv
作る	成果物/D3-4/の要求一覧とレビュー
作る	成果物/D3-4/index.html（試作品）

Copyright © Givery, Inc. All rights reserved

ワーク

D3-4の流れ

▶ 103秒の動画で見る

工夫はAIを開く前に自分で1つ決めます

■ 自分の手で　■ AIと一緒に　□ 座学・理解度チェック

演習の枠 [40min]

STEP1 [5min]	STEP2 [20min]	STEP3 [8min]	STEP4 [7min]

■ STEP1	メモと工夫	ヒアリングメモを読み、 自分で加える工夫を1つ決めます
■ STEP2	要求抽出から試作品まで	要求抽出のコマンドで要求一覧とレビューを作り、 指摘を自分で判断してから試作品のコマンドを送ります
■ STEP3	ブラウザで確かめる	試作品を開き、 条件4つと自分の工夫を操作して確かめます
■ STEP4	チャットに貼る	画面のスクリーンショットと工夫の1行を、 Zoomのチャットに貼ります

Copyright © Givery, Inc. All rights reserved

条件4つの確かめ方

4つの条件はブラウザで操作して1つずつ確かめます

条件	操作	満たしている状態
予約の一覧	基準日を12月1日にする	その日と翌日の予約が出る
会議室と日付での絞り込み	会議室を1つと、日付を1日選ぶ	その日の予約と、 空いている時間が出る
重複予約のエラー表示	一覧にある予約と同じ会議室、 同じ時間帯で予約を足す	エラーが出て、 一覧の件数が増えない
予約の取り消し	予約を1件取り消す	一覧からその予約が消える

満たしていない条件は、言葉で直させます

どの操作で何が起きたかを1文で伝えてください。
直したら、4つ全部をもう一度操作します。

Copyright © Givery, Inc. All rights reserved

試作品の開き方

▶ 100秒の動画で見る

Day1と同じくpython3のサーバーで配ってから開きます

Ubuntu22.04のターミナル

```
cd ~/idcc-day3
python3 -m http.server 8000
```

Windowsのブラウザで開くURL

```
http://localhost:8000/成果物/D3-4/index.html
```

場面	すること
サーバーを動かす場所	VS Codeのターミナルの右上の［＋］で2つ目を開き、そこで動かす
サーバーを止める	そのターミナルでCtrl+Cを押す
画面が古いまま	ブラウザで再読み込みする（Ctrl+F5）
データが出ない	ファイルをダブルクリックで開いていないかを見る。file://ではデータを読めない

試作品のコマンドが返す報告にも、開き方の1行が書かれます。

Copyright © Givery, Inc. All rights reserved

スクリーンショットとチャットの1行

画面1枚と工夫を書いた1行をチャットに貼ります

1	**撮る**	試作品の画面を開いたまま、Windows＋Shift＋Sで範囲を選んで撮る
2	**貼る**	zoom Zoomのチャット欄に貼り付ける
3	**1行を添える**	「D3-4の工夫は<自分で足したこと>です」の形で書いて送る

貼るのは架空の題材の画面だけです

自分の業務の画面やファイルは貼りません。
講師がチャットから数枚を選び、工夫を言葉で紹介します。発表はありません。

Copyright © Givery, Inc. All rights reserved

発展

D3-4早く終わった方へ

追加と考察

条件4つのうち、最初の1回で満たさなかった条件と、
直すときに送った指示を1行ずつ残します。

発展課題

D3-4+　~/.claude/CLAUDE.mdに自分の好みを書き、もう一度作らせて雰囲気が変わるかを見ます。
試したあとは、追記した行を消します。

時間が余った方だけ取り組みます。答え合わせは講師が口頭で行います。

Copyright © Givery, Inc. All rights reserved

3日間のまとめ

3日間でAIエージェントの基本から**ハーネスの作り方**までを通しました

回	身に付くこと	作ったもの
Day1	対話型AIとAIエージェントの違いを知り、言葉だけで 動く画面を作り、注意点を守って確かめる	工数集計の画面1枚
Day2	CLAUDE.mdと4種類のハーネスとコンテキストが、 同じ依頼の結果をどう変えるかを自分の画面で見る	見比べ表5枚と、 自分で育てたCLAUDE.md
Day3	ハーネスの全体像を知り、成功した手順をSkillに 落とし、別のテーマに使い回す	自作のSkill 2本と、 会議室予約の試作品

3日間の柱は、エージェントの基本とハーネスとコンテキストの3つです。

▶ 84秒の動画で見る

Copyright © Givery, Inc. All rights reserved

持ち帰るもの

研修のあとは配布フォルダと自分で作ったファイルを持ち帰ります

持ち帰るもの	どこにあるか	現場での使い方
自分で育てたCLAUDE.md	Day2の配布フォルダ	自分のチームの規約に書き換える
自作のSkill 2本	idcc-day3/.claude/skills/	自分のプロジェクトの.claude/skills/に写す
理想的なハーネス一式	idcc-day3/（配布ZIPと同じ）	SubagentとCommandの組み方の手本にする
ハーネスの作り方の手順	教材サイトのハンズオンガイド	公式の雛形、成功体験、skill-creatorの3つ

> **研修用のアカウントは研修のあと使えません**
>
> どのファイルも文字だけなので、社内で使えるClaude Codeの環境にそのまま写して使えます。
> 研修の端末から持ち出してよいかは、社内の決まりに従ってください。

Copyright © Givery, Inc. All rights reserved

現場に戻ってからの最初の一歩

まず使える環境を確かめ、小さなSkillを1つ作ります

1. **使える環境を確かめる**
   社内で使えるClaude Codeの契約と、AIに入れてよい情報の決まりを確かめる

2. **CLAUDE.mdを1枚書く**
   持ち帰ったCLAUDE.mdを、自分のチームの規約と応答の長さに書き換える

3. **Skillを1つ作る**
   毎回貼っている頼み方を1つ選び、D3-2の手順でSkillにする

止めたい操作が決まっていれば、4つ目としてHookを1つ足します。

Copyright © Givery, Inc. All rights reserved

クイズ

理解度チェック

必ず止めたい操作を置く場所はどれですか

A　CLAUDE.mdに、しないこととして書く

B　Skillの手順の最後に書く

C　settings.jsonのHook（PreToolUse）で止める

D　Subagentのdescriptionに書く

答えはこの組の問題がそろってから確かめます。チャットに記号だけ書いてください。

Copyright © Givery, Inc. All rights reserved

クイズ

理解度チェック 続き

どのフォルダでも使う自分の口調の指定はどこに書きますか

A ~/.claude/CLAUDE.md

B プロジェクトのCLAUDE.md

C プロジェクトの.claude/settings.json

D プロジェクトの.claude/skills/

答えはこの組の問題がそろってから確かめます。チャットに記号だけ書いてください。

Copyright © Givery, Inc. All rights reserved

クイズ

理解度チェック 続き

生成とレビューを別のSubagentに分ける理由はどれですか

A　2体で動くと、応答が短くなるから

B　モデルを自動で切り替えられるから

C　レビュー役がファイルを書けなくなるから

D　書いた経緯を持たない別のコンテキストで点検できるから

答えはこの組の問題がそろってから確かめます。チャットに記号だけ書いてください。

Copyright © Givery, Inc. All rights reserved

クイズ

理解度チェックの答え

答えはC・A・Dの順です

問1 **C** **Hookで止める**

CLAUDE.mdは文脈として読まれるだけです。
Hookは決めた瞬間に必ず動きます

問2 **A** **~/.claude/CLAUDE.md**

ユーザーの置き場は、自分が開く全部のフォルダで
読まれます

問3 **D** **別のコンテキストで点検できる**

書いたばかりの内容に引っ張られずに、
結果と基準だけを見られます

Copyright © Givery, Inc. All rights reserved

質疑とアンケート

ご質問はチャットか口頭でどうぞ

まとめのあとに質疑の時間を取ります

アンケートへのご回答をお願いします

URLはチャットでお送りします

教材サイト（ハンズオンガイドとスライドを読み返せます）

https://idcc12.give-app.net

Copyright © Givery, Inc. All rights reserved



株式会社ギブリー

〒150-0036　東京都渋谷区南平台町15-13　帝都渋谷ビル8F

https://givery.co.jp/

3日間のご参加、ありがとうございました。

Copyright © Givery, Inc. All rights reserved