

Day2

AI駆動開発
Claude Code研修

CLAUDE.md、ハーネス、コンテキスト

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18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

> **同じ依頼の結果が、ハーネスと資料で変わることを確かめる**
> **差を生んだファイルを見つけ、自分の現場で書くことを決める**
>
> 講師の画面はハーネスなし、自分の画面はハーネスありで、同じ文面を送って見比べます。

学びを確かめる成果物（idcc-day2の中）

ファイル	確かめられること
見比べ/の見比べ表5枚	演習ごとに、講師の画面との差が2つと、差を生んだファイル名がある
CLAUDE.md	/initで作り、AGENTS.mdの取り込みとコンテキストの参照を足してある

Copyright © Givery, Inc. All rights reserved

3日間の見取り図

Day2で見たハーネスを、Day3で1つのプロジェクトにそろえます

Day1　11月10日（火）

身に付くこと

対話型AIとAIエージェントの
違いを知り、言葉だけで
動く画面を作る

作るもの

工数集計の画面1枚

Day2　11月18日（水）

身に付くこと

**CLAUDE.mdと4種類の
ハーネスとコンテキストで、
同じ依頼の結果が
どう変わるかを見る**

作るもの

見比べ表5枚
自分で育てたCLAUDE.md

Day3　11月25日（水）

身に付くこと

ハーネスの全体像を知り、
成功した手順をSkillにして、
別のテーマに使い回す

作るもの

自作のSkill2本
別テーマの成果物

配布フォルダは回ごとに別です。Day1が終わっていなくても、今日の演習は同じ状態から始められます。

Copyright © Givery, Inc. All rights reserved

本日の進め方

見比べの演習は4本とも同じ進め方です

前半 [80min]

区分	内容	時間
開始	おさらいと配布フォルダの展開	[10min]
座学	CLAUDE.mdと/init、AGENTS.md	[10min]
D2-1	/initとAGENTS.mdの見比べ	[10min]
座学	ハーネスの考え方と4種類の地図	[10min]
D2-2	Subagentsの見比べ	[20min]
D2-3	Skillsの見比べ	[20min]

後半 [90min]

区分	内容	時間
D2-4	Hooksの見比べ	[20min]
デモ	MCP（講師の画面）	[10min]
座学	コンテキストエンジニアリング	[15min]
D2-5	コンテキストの見比べ	[25min]
閉会	まとめ、質疑、アンケート	[20min]

休憩 [10min]は、前半と後半のあいだに取ります

Copyright © Givery, Inc. All rights reserved

Day1のおさらい

Day1で覚えた操作を、今日の見比べでも**そのまま使います**

Day1で使ったもの	今日の使い道
承認の画面で、コマンドを読む	D2-1とD2-2で読むコマンドを許可し、D2-3とD2-5のmvnはEscで断る
書き換えの行を読む	D2-3からD2-5で、どのファイルが書き換わったかを見比べる
`/context`	D2-2で、Messagesの数字を見比べ表に書き写す
`/usage`	利用枠のうち、使った量が気になったときに見る
`/exit`	見比べの前に起動し直し、前の演習の会話を持ち越さない

D2-3からD2-5はaccept edits onで送るので、ファイルの書き換えと、フォルダの中のrmやmvは承認なしで進みます。

Copyright © Givery, Inc. All rights reserved

利用枠の使い方

研修用の利用枠を使うので、同じ文面は1回だけ送ります

1	**応答は要点だけ**	AGENTS.mdに「応答は要点だけ、10行以内」と書いてあります。長い表は、欲しいときだけ頼みます。
2	**同じ文面は1回だけ**	見比べでは、講師がハーネスなしの側を1回、皆さんがハーネスありの側を1回だけ送ります。
3	**Subagentは1体まで**	必須の演習で同時に動くSubagentは1体までにしてあります。
4	**モデルは固定しない**	配布フォルダの設定とSubagentに、モデルを固定する行は書いていません。各自の既定のモデルで動きます。
5	**使った量を見る**	/usageで、セッションの利用量と利用枠の上限を確かめられます。

Copyright © Givery, Inc. All rights reserved

今日の題材は、架空の会社の勤怠・工数管理システムです

サンプル情報サービス株式会社（架空）の社内システムKINTAIです。2014年に稼働し、
JSPとSpring MVCとMyBatisで作られています。

画面ID	画面名	中身	今日の演習
KIN010	メニュー	各画面へのリンク、ログイン中の氏名と締め状態	
KIN020	日次勤怠入力	勤務区分、出勤、退勤、休憩、実働	
KIN030	工数入力	日付ごとのプロジェクト、作業区分、作業時間	D2-2、D2-3、D2-4
KIN040	交通費申請	日付、区間、金額、目的	
KIN050	月次締め・CSV	締め状態、CSV出力、パートナー実績の取込	
PJ010	収支ダッシュボード	新設する画面	D2-5

今日は画面を動かさず、Claude Codeの動きと差分で見比べます。画面を動かすのは講師のデモと発展だけです。

Copyright © Givery, Inc. All rights reserved

Day2の配布フォルダ

ハーネスは.claude/とAGENTS.mdに入っています

idcc-day2/	
AGENTS.md	常に読まれる約束ごと（D2-1）
はじめにお読みください.md	架空の題材であることと、使い方
（KINTAIのプロジェクト）	Java、JSP、MyBatisのMapper、メッセージ
docs/	改修依頼書、画面設計書、規約（D2-5）
.claude/	ハーネスの置き場
agents/	Subagentの定義（D2-2）
skills/	Skillのフォルダ（D2-3）
settings.json	Hooksの設定（D2-4）
context/	AIに常に守らせる要点（D2-5）
見比べ/	皆さんが書き込む見比べ表5枚

- 青い名前が、今日の演習で効くはずのファイルとフォルダです

Copyright © Givery, Inc. All rights reserved

ハーネスの置き場の地図

演習ごとに、効くはずのファイルが決まっています

演習	効くはずのファイル	種類	読まれ方と動き方
D2-1	AGENTS.mdとCLAUDE.md	指示ファイル	常に読まれる
D2-2	`.claude/agents/spec-reviewer.md`	Subagent	別のコンテキストで動く
D2-3	`.claude/skills/kintai-validation/`	Skill	依頼が合うと読まれる
D2-4	.claude/settings.jsonのhooks	Hook	決まった瞬間に必ず動く
D2-5	docs/と.claude/context/	AIに渡す資料	要点は常に、資料は作業ごと
デモ	講師の画面のMCPの設定	MCP	外の道具につなぐ

D2-1のCLAUDE.mdは、/initで皆さんが作ります。MCPは配布フォルダに入っていません。

Copyright © Givery, Inc. All rights reserved

展開してからVS Codeで開き、claudeを起動します

1	**ZIPを受け取る**	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。
2	**展開する**	Ubuntuのターミナルでunzipを使い、ホームにidcc-day2を展開します。
3	**VS Codeで開く**	VS CodeをWSLにつなぎ、~/idcc-day2のフォルダを開きます。
4	**起動する**	VS Codeの統合ターミナルでclaudeを起動します。

AGENTS.mdが読まれているかは、入力欄に/memoryと打つと一覧で確かめられます。
入力欄の下にmanual mode onと出ていれば、書き換えの前に承認の画面が出ます。
コマンドの全文の置き場所は、ハンズオンガイドの「始める前の準備」です。

出典　https://code.claude.com/docs/en/memory#agents-md（2026年10月21日閲覧）

Copyright © Givery, Inc. All rights reserved

指示ファイルの読み込まれ方

CLAUDE.mdは、会話を始めるたびに全文が読み込まれます

置き場	効く範囲	読み込む時点
`~/.claude/CLAUDE.md`	自分が開く全部のフォルダ	起動したとき
`./CLAUDE.md`	このプロジェクトを開いた全員	起動したとき
`./CLAUDE.local.md`	このプロジェクトの自分だけ	起動したとき
サブフォルダのCLAUDE.md	そのフォルダの中	そこのファイルを読んだとき

どのファイルも上書きせず、範囲の広い順につなげて読み込みます。プロジェクトのCLAUDE.mdは./.claude/CLAUDE.mdに置いても同じです。

読み込まれたファイルは、入力欄に/memoryと打つと一覧で出ます。

出典　https://code.claude.com/docs/en/memory（2026年10月21日閲覧）

▶ 27秒の動画で見る

Copyright © Givery, Inc. All rights reserved

AGENTS.mdの今の扱い

CLAUDE.mdがあると、AGENTS.mdは直接は読まれません

作業フォルダとその上にあるファイル	Claude Codeが読むもの
AGENTS.mdだけがある	AGENTS.md
AGENTS.mdとCLAUDE.mdがある	CLAUDE.mdだけ
CLAUDE.mdが @AGENTS.md で取り込んでいる	CLAUDE.mdと、取り込んだAGENTS.md

CLAUDE.mdの先頭に書く取り込み

```
@AGENTS.md

## Claude Code
Claude Codeだけに伝えることを書く
```

この判定に数えるのはCLAUDE.md、.claude/CLAUDE.md、CLAUDE.local.mdです。~/.claude/CLAUDE.mdは数えません。

AGENTS.mdを直接読むのはv2.1.277以降です。

▶ 95秒の動画で見る

出典　https://code.claude.com/docs/en/memory#agents-md（2026年10月21日閲覧）

Copyright © Givery, Inc. All rights reserved

/initで作られるもの

▶ 95秒の動画で見る

/initはプロジェクトを読み、CLAUDE.mdの最初の版を作ります

/initが書くもの

- コードを読んで見つけた、ビルドとテストのコマンド
- プロジェクトの規約と構成
- CLAUDE.mdがすでにあるときは、上書きせずに改善案を出す

人が足すもの

- コードからは読み取れない決まり
- AGENTS.mdを取り込む行（D2-1）
- AIに常に守らせる資料への参照（D2-5）

Refine from there with instructions Claude wouldn't discover on its own.

/initのあとは、Claudeが自分では見つけられない指示を書き足していきます。

出典　https://code.claude.com/docs/en/memory（2026年10月21日閲覧）

Copyright © Givery, Inc. All rights reserved

CLAUDE.mdに書くもの

毎回守らせたい短い決まりだけを、200行以内で書きます

CLAUDE.mdに書くもの	別の置き場へ回すもの	回す先
ビルドとテストのコマンド	何段もある手順	Skill（.claude/skills/）
コーディング規約と命名	一部のフォルダだけの決まり	.claude/rules/かサブフォルダのCLAUDE.md
フォルダの構成	必ず止めたい操作	Hook（.claude/settings.json）
いつも守る短い決まり	長い設計書	docs/に置き、作業のときに @ で渡す

公式ドキュメントは、CLAUDE.mdを1ファイル200行以内にするよう勧めています。長いほど窓を使い、
指示が守られにくくなります。

出典　https://code.claude.com/docs/en/memory（2026年10月21日閲覧）
https://code.claude.com/docs/en/features-overview

▶ 95秒の動画で見る

Copyright © Givery, Inc. All rights reserved

D2-1

/initとAGENTS.mdの見比べ

[10min]

作るもの

CLAUDE.md、見比べ表（D2-1）

Copyright © Givery, Inc. All rights reserved

ワーク

演習D2-1のねらいと目安

ねらい

/initでCLAUDE.mdを作り、配布されたAGENTS.mdと読み比べます。2つのファイルの役割の違いと、
取り込む行の書き方をつかみます。

できあがりの目安

- [ ] CLAUDE.mdがあり、AGENTS.mdを取り込む行が入っている
- [ ] 見比べ表（D2-1）に、2つのファイルの役割の違いが1行ずつある
- [ ] 起動し直したあとの/memoryに、AGENTS.mdが @-imported で並ぶ

使うファイルと書くファイル

区分	ファイル	使い方
使う	AGENTS.md	/initの前に、配布された約束ごとを読む
作る	CLAUDE.md	/initで作り、取り込む行を確かめる
書く	見比べ/D2-1_CLAUDE.mdとAGENTS.md.md	2つのファイルの役割を1行ずつ書く

Copyright © Givery, Inc. All rights reserved

ワーク

D2-1の流れ

/initの結果を読み、取り込む行があるかを確かめます

1	**確かめる**	AGENTS.mdを読み、入力欄に/memoryと打って、読み込まれているファイルを見ます。
2	**送る**	/initを送ります。Claudeがプロジェクトを読み、CLAUDE.mdを書きます。
3	**見比べる**	CLAUDE.mdに @AGENTS.md の行があるかを見ます。無ければ見出しのすぐ下に足し、 見比べ表に役割を1行ずつ書きます。
4	**起動し直す**	/exitのあとにclaudeで起動し直し、/memoryにAGENTS.mdが @-imported で並ぶかを見ます。

■ 自分の手で　■ AIと一緒に

この演習だけは講師の画面と比べず、自分の2つのファイルを見比べます。STEPの分数はハンズオンガイドのとおりです。

Copyright © Givery, Inc. All rights reserved

解説

D2-1で見るところ

2つのファイルに同じ決まりが重なっていないかを見ます

- [ ] /initが書いたビルドとテストのコマンド、フォルダの構成
- [ ] AGENTS.mdにだけある決まり（応答の長さなど）
- [ ] 取り込む行の位置と、2つのファイルで重なっている決まり
- [ ] 起動し直したあとの/memoryで、@-importedと並ぶファイル

> CLAUDE.mdがAGENTS.mdを取り込んでいないと、次の起動からAGENTS.mdの約束ごとは読まれません。

出典　https://code.claude.com/docs/en/memory#agents-md（2026年10月21日閲覧）

Copyright © Givery, Inc. All rights reserved

ハーネスとは

ハーネスは、AIの周りに置いて動きを決めるファイルの一式です



公式ドキュメントは、これらをClaude Codeの拡張の仕組みとして並べています。

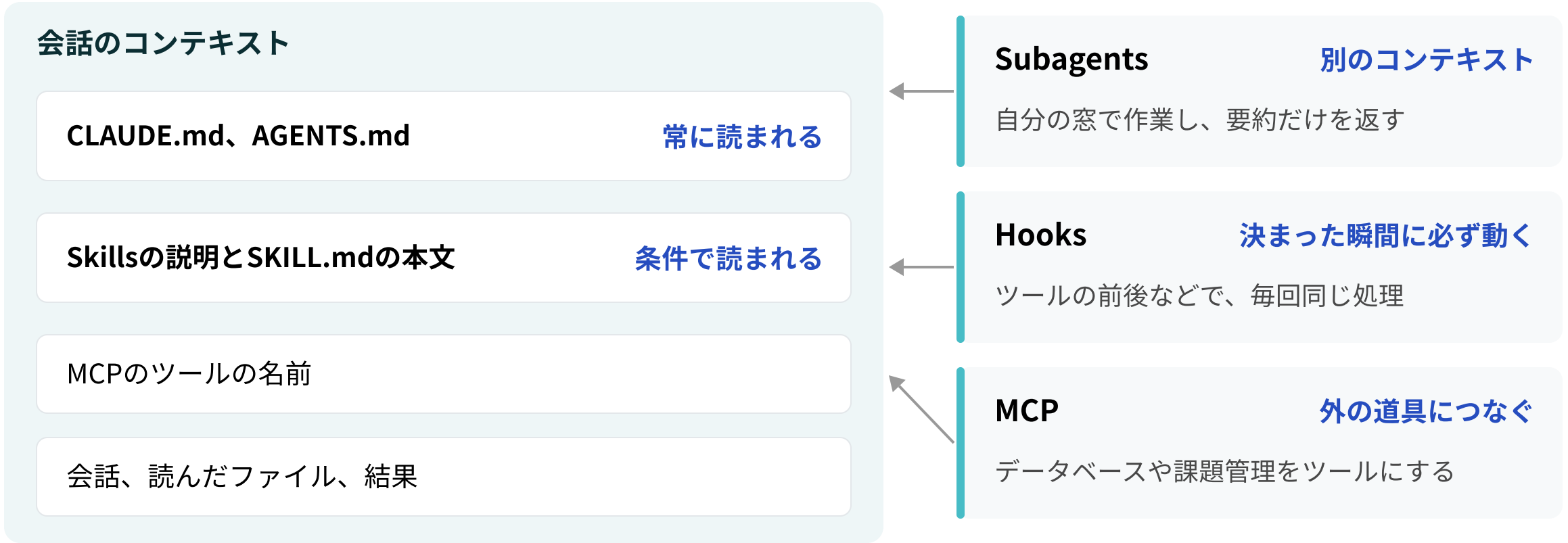
出典　https://code.claude.com/docs/en/features-overview（2026年10月21日閲覧）

▶ 27秒の動画で見る

Copyright © Givery, Inc. All rights reserved

4種類の地図

4種類は、読まれる時点と動く場所が違います



矢印は、Subagentの要約やHookとMCPの結果が会話のコンテキストに戻る道筋です。

Copyright © Givery, Inc. All rights reserved

読み込まれる時点と量

▶ 96秒の動画で見る

常に読まれるものほど、毎回の依頼で窓を使います

仕組み	読み込む時点	読み込むもの	窓の使い方
CLAUDE.md	会話の開始時	全文	毎回の依頼に入る
Skills	開始時と、使うとき	開始時は説明、使うときに本文	説明だけなら少ない
MCP	会話の開始時	開始時は名前、使うときに定義	使うまでは少ない
Subagents	呼ばれたとき	新しい別のコンテキスト	親の会話とは別
Hooks	決まった時点	何も読み込まない	結果を返したときだけ

表は公式ドキュメントのContext cost by featureを訳したものです。

出典　https://code.claude.com/docs/en/features-overview#understand-context-costs（2026年10月21日閲覧）

Copyright © Givery, Inc. All rights reserved

見比べ演習の進め方

講師はハーネスなし、皆さんはハーネスありで同じ文面を送ります

講師の画面

ハーネスなし

Zoomで共有します

同じ文面を
同時に送る

自分の画面

ハーネスあり

idcc-day2をそのまま使います

STEP 1	STEP 2	STEP 3	STEP 4
予想	同時に送る	見比べ	現場
[3min]	[8min]	[6min]	[3min]

見比べ表はidcc-day2/見比べ/に5枚あります。D2-5だけはSTEP 1でCLAUDE.mdに参照を足すので[25min]です。

講師はD2-2からD2-4で、.claude/の中身とCLAUDE.mdを抜いたフォルダを使います。D2-5では、
さらにdocs/と.claude/context/も抜きます。

Copyright © Givery, Inc. All rights reserved

見比べのときの約束

差を見つけたら、必ずファイル名まで戻ります

1	**講師の一言**	講師は送る前に毎回「この画面にはハーネスがない」と言います。
2	**送るのは1回**	文面は講師の合図で1回だけ送ります。送り直しはしません。
3	**待つ間に読む**	Claudeが動いている間に、効くはずのファイルを開いて読みます。
4	**差が出ないとき**	出力は毎回ばらつきます。差が出た方の画面を講師が紹介し、ファイルの行に戻ります。
5	**Zoomを前に**	講師の画面を見る間は、VS Codeを最小化してZoomの画面を前に出します。

Copyright © Givery, Inc. All rights reserved

D2-2

Subagentsの見比べ

[20min]

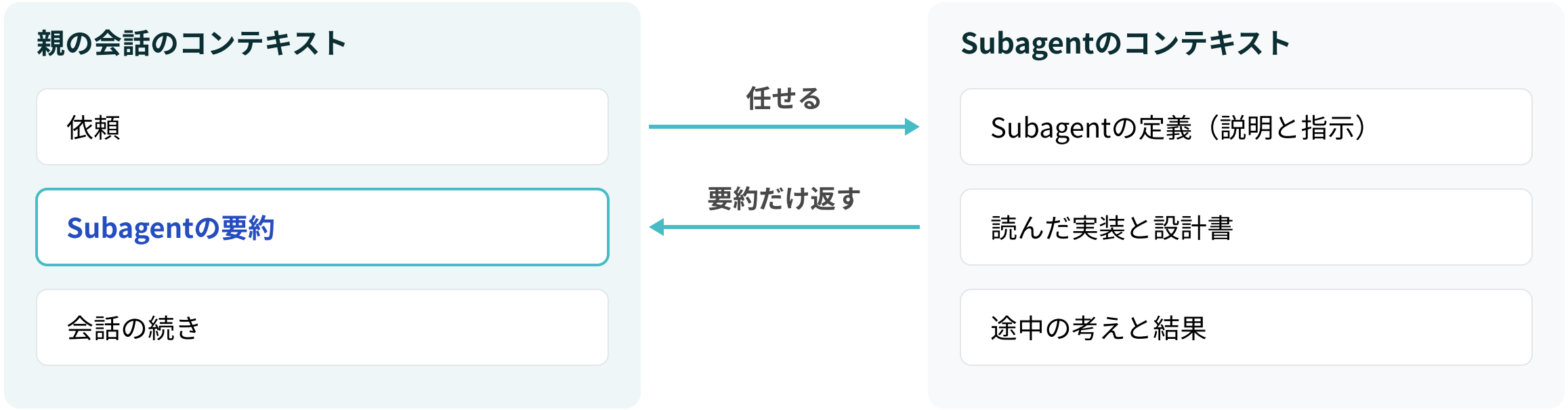
作るもの

見比べ表（D2-2）

Copyright © Givery, Inc. All rights reserved

Subagentsとは

Subagentは別のコンテキストで作業し、要約だけを返します



大量に読む作業を親の会話の外で行い、結論だけを受け取る仕組みです。

観点や使える道具を絞った担当を作るときにも使います。

▶ 29秒の動画で見る

出典　https://code.claude.com/docs/en/sub-agents（2026年10月21日閲覧）

Copyright © Givery, Inc. All rights reserved

Subagentの置き場と形

Subagentは.claude/agents/に1体1ファイルで置きます

一般形の例（.claude/agents/spec-reviewer.md）

```
---
name: spec-reviewer
description: 実装を設計書と突き合わせて確かめる
tools: Read, Grep, Glob
---
あなたは設計書との突き合わせを
担当するレビュアーです。
（観点と、返す表の形を書く）
```

項目	中身
name	必ず書く呼び名
description	Claudeが任せるかを決める説明
tools	使える道具を絞る
model	書かなければ会話と同じモデル
本文	Subagentへの指示

置き場は.claude/agents/（このプロジェクト）と~/.claude/agents/（自分の全プロジェクト）です。
研修のファイルにはmodelの行を書いていません。

▶ 96秒の動画で見る

出典　https://code.claude.com/docs/en/sub-agents（2026年10月21日閲覧）

Copyright © Givery, Inc. All rights reserved

Subagentと利用量

親の会話は軽くなりますが、Subagentも利用枠を使います

親の会話で読ませた場合

親の窓に入るもの
読んだファイルと途中の考えの全部

利用枠に数えるもの
親の会話の依頼

Subagentに任せた場合

親の窓に入るもの
Subagentの要約だけ

利用枠に数えるもの
親の会話の依頼と、Subagent自身の依頼

> It also sends its own requests, which count toward the same usage limits as your main conversation.
>
> Subagentは自分でも依頼を送り、その分も親の会話と同じ利用枠に数えられます。

研修の必須の演習で、同時に動くSubagentは1体までです。

出典　https://code.claude.com/docs/en/sub-agents（2026年10月21日閲覧）
https://code.claude.com/docs/en/costs

▶ 94秒の動画で見る

Copyright © Givery, Inc. All rights reserved

ワーク

演習D2-2のねらいと目安

ねらい

講師の画面と自分の画面で、同じレビューの依頼を送ります。Subagentが別のコンテキストで動くと、
親の会話に何が残るかを見比べます。

できあがりの目安

- [ ] 見比べ表（D2-2）に、講師の画面との差が2つと、差を生んだファイル名がある
- [ ] 自分の/contextの数字を、見比べ表に書き写してある

使うファイルと書くファイル

区分	ファイル	使い方
使う	`.claude/agents/spec-reviewer.md`	STEP 1とSTEP 2で中身を読む
使う	docs/設計/画面設計書_KIN030_工数入力_改修.md	レビューで突き合わせる設計書
書く	見比べ/D2-2_Subagents.md	差を2つと、/contextの数字を書く

Copyright © Givery, Inc. All rights reserved

ワーク

D2-2の流れ

STEP 1の3分は、依頼を送らずに予想を書きます

■ 自分の手で　■ AIと一緒に　　演習の枠 [20min]

STEP 1 [3min]	STEP 2 [8min]	STEP 3 [6min]	STEP 4 [3min]

■ STEP 1	予想	spec-reviewer.mdを読み、/contextのCustom agentsの欄でspec-reviewerを確かめます。予想を1行で書きます。
■ STEP 2	同時に送る	manual mode onのまま、講師の合図で同じ文面を送ります。 待つ間に、spec-reviewer.mdの手順を読みます。
■ STEP 3	見比べ	差を2つと、差を生んだファイル名を書きます。/contextの数字も書き写します。
■ STEP 4	現場での置き換え	自分の現場なら、Subagentに何を任せるかを1行で書きます。

送る文面の全文は、ハンズオンガイドのD2-2にあります。

Copyright © Givery, Inc. All rights reserved

解説

D2-2で見るところと置き換え

差は、/contextの数字と親の会話に残ったもので見ます

見比べで見るところ（STEP 3）

1. レビューがどこで動いたかを示す画面の表示
2. 親の会話に残ったものの量と形
3. /contextの数字（自分の画面と講師の画面）
4. 指摘が設計書のどの項目を根拠にしているか

自分の現場なら（STEP 4）

大量に読んで、結論だけが欲しい作業がSubagentに向きます。

- 設計書と実装の突き合わせレビュー
- テストやビルドのログの読み込み
- 改修の影響範囲の調べもの

差を見つけたら、spec-reviewer.mdのどの行が効いたかに戻ります。

Copyright © Givery, Inc. All rights reserved

D2-3

Skillsの見比べ

[20min]

作るもの

見比べ表（D2-3）

Copyright © Givery, Inc. All rights reserved

Skillsとは

Skillは説明だけを常に読み、本文は使うときに読み込みます

会話の開始時

kintai-validationの説明

ほかのSkillの説明

説明だけが窓に入ります

依頼が説明に合ったとき

SKILL.mdの本文

手順、置き場、書き方

本文が窓に入り、その手順で進めます

/kintai-validationと名前で呼ぶこともできます

カスタムコマンド（.claude/commands/）はSkillにまとめられ、
今も同じように動きます。

出典　https://code.claude.com/docs/en/skills（2026年10月21日閲覧）
https://code.claude.com/docs/en/features-overview

▶ 28秒の動画で見る

Copyright © Givery, Inc. All rights reserved

Skillの置き場と形

Skillは1つのフォルダに、SKILL.mdと補助のファイルを置きます

```
.claude/skills/
└ kintai-validation/
    ├ SKILL.md              説明と手順。必須
    └ （補助のファイル）      必要なときだけ読む
```

置き場	使える範囲
.claude/skills/<名前>/SKILL.md	プロジェクト
~/.claude/skills/<名前>/SKILL.md	全プロジェクト

フォルダ名かnameが、/で呼ぶときの名前になります。

出典　https://code.claude.com/docs/en/skills（2026年10月21日閲覧）

一般形の例（SKILL.md）

```
---
name: kintai-validation
description: 入力チェックを足すときの手順
---
## 手順
1. チェックの種類を決める
2. メッセージの置き場を決める
3. 決めた場所に書き足す
```

▶ 95秒の動画で見る

Copyright © Givery, Inc. All rights reserved

ワーク

演習D2-3のねらいと目安

ねらい

同じ入力チェックの依頼を送り、Skillがあるときとないときで、
Claudeが書き足す場所と書き方がどう変わるかを見比べます。

できあがりの目安

- [ ] 見比べ表（D2-3）に、講師の画面との差が2つと、差を生んだファイル名がある
- [ ] 自分の画面の書き換えにE-KIN030-003がある

使うファイルと書くファイル

区分	ファイル	使い方
使う	`.claude/skills/kintai-validation/SKILL.md`	STEP 1とSTEP 2で手順を読む
確かめる	書き換わったKIN030のファイル	書き換えの行で、書き換わった場所を見る
書く	見比べ/D2-3_Skills.md	差を2つと、差を生んだファイル名を書く

Copyright © Givery, Inc. All rights reserved

ワーク

D2-3の流れ

STEP 2はaccept edits onで送り、mvnの承認はEscで断ります

■ 自分の手で　■ AIと一緒に　　演習の枠 [20min]

STEP 1 [3min]	STEP 2 [8min]	STEP 3 [6min]	STEP 4 [3min]

■ STEP 1	予想	/skillsでkintai-validationを確かめ、SKILL.mdの表を読みます。 予想を1行で書きます。
■ STEP 2	同時に送る	accept edits onにして、講師の合図で同じ文面を送ります。 mvnの承認が出たらEscで断ります。
■ STEP 3	見比べ	書き換えの行と書き換わったファイルを読み、講師の画面との差を2つと、 差を生んだファイル名を書きます。
■ STEP 4	現場での置き換え	自分の現場なら、どの手順をSkillにするかを1行で書きます。

送る文面の全文は、ハンズオンガイドのD2-3にあります。

Copyright © Givery, Inc. All rights reserved

解説

D2-3で見るところと置き換え

差は、書き換わったファイルの場所とメッセージで見ます

見比べで見るところ（STEP 3）

1. 書き換わったファイルの数と場所
2. メッセージIDと、メッセージを書いた場所
3. Skillが読み込まれたことを示す画面の表示
4. 同じ種類のチェックの書き方が、
   2つの画面でそろっているか

自分の現場なら（STEP 4）

**繰り返す手順と、ときどき要る参照資料が
Skillに向きます。**

- 入力チェックとメッセージの足し方
- 帳票やCSVの出力の書式
- 障害報告の書き方

毎回守る短い決まりはCLAUDE.mdに、ときどき使う手順はSkillに置きます。

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

D2-4

Hooksの見比べ

[20min]

作るもの

見比べ表（D2-4）

Copyright © Givery, Inc. All rights reserved

Hooksとは

Hookは、決まった時点でClaudeの判断を待たずに動きます

CLAUDE.mdやSkillに書いた禁止

Claudeへの依頼として読まれます。
Claudeが判断するので、
守られないことがあります。

Hook

決まった時点でスクリプトが動きます。
止める処理を書けば、
毎回その時点で止まります。

> Claude treats them as context, not enforced configuration.
> To block an action regardless of what Claude decides, use a PreToolUse hook instead.
>
> ClaudeはCLAUDE.mdを文脈として扱います。必ず止めるには、PreToolUseのHookを使います。

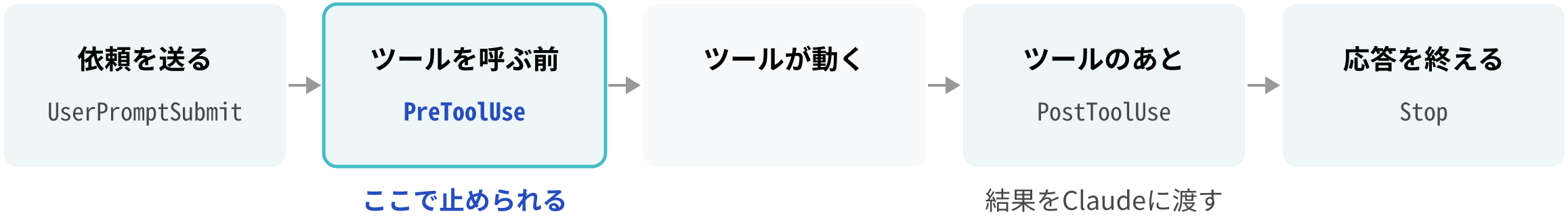
出典　https://code.claude.com/docs/en/memory（2026年10月21日閲覧）
https://code.claude.com/docs/en/features-overview

▶ 91秒の動画で見る

Copyright © Givery, Inc. All rights reserved

Hookが動く時点

PreToolUseのHookは、ツールが動く前に操作を止められます



止め方

Hookのスクリプトが終了コード2で終わると、ツールは動きません。標準エラーに書いた理由がClaudeに渡り、Claudeはやり方を変えます。どのツールで動かすかはmatcherで絞ります。

出典　https://code.claude.com/docs/en/hooks-guide（2026年10月21日閲覧）

▶ 32秒の動画で見る

Copyright © Givery, Inc. All rights reserved

Hookの置き場と形

HookはSettings.jsonのhooksに、時点と処理を書きます

一般形の例（.claude/settings.json）

```
{
  "hooks": {
    "PreToolUse": [
      {
        "matcher": "Edit|Write",
        "hooks": [
          { "type": "command",
            "command": "python3 <止めるスクリプト>" }
        ]
      }
    ]
  }
}
```

出典　https://code.claude.com/docs/en/hooks-guide（2026年10月21日閲覧）

置き場と効く範囲

~/.claude/settings.json
自分の全プロジェクト

.claude/settings.json
このプロジェクトの全員

.claude/settings.local.json
このプロジェクトの自分だけ

研修のHookのスクリプトは、
python3の標準ライブラリだけで
書いてあります。

▶ 97秒の動画で見る

Copyright © Givery, Inc. All rights reserved

ワーク

演習D2-4のねらいと目安

ねらい

設計書を書き換える依頼を送り、Hookがある画面で何が止まり、どんな理由が表示されるかを見比べます。

できあがりの目安

- [ ] Hookが止めた表示が、自分の画面に出ている
- [ ] docs/の設計書が変わっていない
- [ ] 見比べ表（D2-4）に、講師の画面との差が2つある

使うファイルと書くファイル

区分	ファイル	使い方
使う	`.claude/hooks/guard_docs.py`	STEP 1で先頭の説明を読む
使う	`.claude/settings.json`	STEP 2の待ち時間にhooksの設定を読む
確かめる	docs/設計/画面設計書_KIN030_工数入力_改修.md	STEP 3で変わっていないことを見る
書く	見比べ/D2-4_Hooks.md	差を2つと、差を生んだファイル名を書く

Copyright © Givery, Inc. All rights reserved

ワーク

D2-4の流れ

提案だけが返ったら、試しの文面を続けて送ります

■ 自分の手で　■ AIと一緒に　　演習の枠 [20min]

STEP 1 [3min]	STEP 2 [8min]	STEP 3 [6min]	STEP 4 [3min]

■	**STEP 1　予想**	/hooksでPreToolUseのHookを開き、guard_docs.pyの説明を読みます。 予想を1行で書きます。
■	**STEP 2　同時に送る**	accept edits onにしてから、講師の合図で同じ文面を送ります。止まったら、 表示された理由を読みます。
■	**STEP 3　見比べ**	設計書が変わったかを開いて確かめ、講師の画面との差を2つ書きます。
■	**STEP 4　現場での置き換え**	自分の現場なら、どの操作をHookで止めるかを1行で書きます。

送る文面の全文は、ハンズオンガイドのD2-4にあります。

Copyright © Givery, Inc. All rights reserved

解説

D2-4で見るところと置き換え

差は、止まった操作と表示された理由で見ます

見比べで見るところ（STEP 3）

1. docs/の設計書が書き換わったか
2. 止めた理由として表示された文
3. ファイルを書き換えたあとに表示されるもの
4. 止められたあとに、Claudeが次にとった行動

自分の現場なら（STEP 4）

毎回同じに動かしたい処理と、
必ず止めたい操作がHookに向きます。

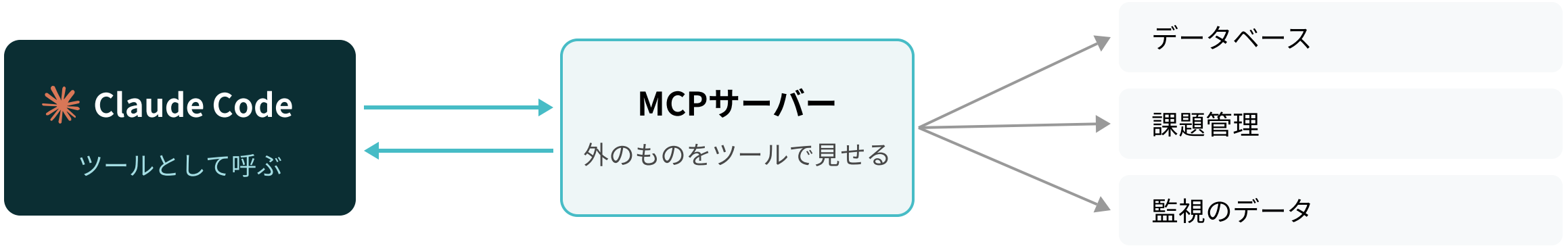
- 承認済みの設計書を書き換えさせない
- 書き換えのたびにリンターを動かす
- 危険なコマンドを実行の前に止める

差を見つけたら、settings.jsonのhooksのどの設定が動いたかに戻ります。

Copyright © Givery, Inc. All rights reserved

MCPとは

MCPは、外のデータや道具をClaudeのツールとしてつなぐ決まりです



会話の開始時に読み込むのは、ツールの名前です。詳しい定義は、Claudeがそのツールを使うときに読み込みます。

Claude Codeの公式ドキュメントは、データベースへの問い合わせや、
課題管理の内容からの実装を例に挙げています。

出典　https://code.claude.com/docs/en/mcp（2026年10月21日閲覧）
https://code.claude.com/docs/en/features-overview

▶ 30秒の動画で見る

Copyright © Givery, Inc. All rights reserved

実演

MCPのデモの構成

▶ 31秒の動画で見る

講師の画面だけで、読み取り専用のデータベースにつなぎます

講師が聞くこと

9月に1日の合計が24時間を超えた社員は誰か

講師のClaude Code
MCPのツールを呼ぶ

→

MCPサーバー
読み取り専用でつなぐ

→

KINTAIのデータ
架空の9月分のCSVの写し

デモで見るところ

- [ ] どのツールが、どんな問い合わせで呼ばれたか
- [ ] 許可を求める画面に何が表示されるか
- [ ] 返ってきた答えを、人がどう確かめるか

皆さんのPCではMCPをつなぎません。仕組みと権限の決め方だけを見ます。

Copyright © Givery, Inc. All rights reserved

MCPの権限の決め方

つなぐ前に、読める範囲と許可の出し方を決めます

決めること	決め方
接続する利用者	読み取り専用の利用者でつなぎます。Claudeが送る問い合わせでデータが変わりません。
置き場	自分だけならローカル（~/.claude.json）、チームで共有するなら.mcp.jsonに置きます。 .mcp.jsonのサーバーは、使う前に承認を求められます。
許可のルール	settings.jsonのpermissionsに、mcp__サーバー名__ツール名 の形で許可と拒否を 書きます。
信頼	信頼できるサーバーだけをつなぎます。外の文章を取ってくるサーバーは、 プロンプトインジェクションの入口になります。

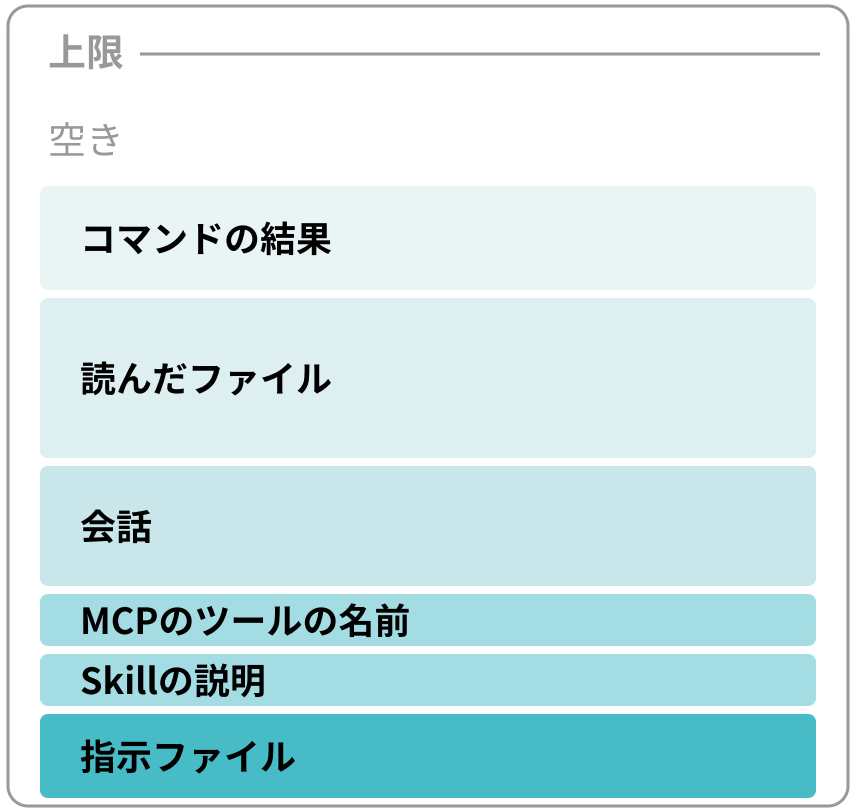
出典　https://code.claude.com/docs/en/mcp（2026年10月21日閲覧）
https://code.claude.com/docs/en/permissions

▶ 31秒の動画で見る

Copyright © Givery, Inc. All rights reserved

コンテキストの窓

窓が埋まるほど、Claudeは前の指示を守りにくくなります



> Claude's context window fills up fast,
> and performance degrades as it fills.
>
> 窓はすぐに埋まり、埋まるほどClaudeの働きは落ちます。前の指示を忘れたり、誤りが増えたりすることがあります。

/contextで、今の窓の使われ方を見られます。

▶ 30秒の動画で見る

出典　https://code.claude.com/docs/en/best-practices（2026年10月21日閲覧）
https://code.claude.com/docs/en/context-window

Copyright © Givery, Inc. All rights reserved

窓に何を入れて何を外すかを、作業の前に決めます

入れ方	入れるもの	今日のファイルと操作
常に入れる	毎回守らせたい短い要点	CLAUDE.md、.claude/context/
作業ごとに渡す	その作業で読む正式な資料	docs/の設計書を @ で指定する
外に出す	大量に読む作業、前の作業の会話	Subagentに任せる、/clearで消す

D2-5では、3つの入れ方をそろえた自分の画面と資料のない講師の画面で、
同じ依頼を比べます。

▶ 29秒の動画で見る

Copyright © Givery, Inc. All rights reserved

AIに渡す資料の一覧

D2-5の配布フォルダには、この9種類が置いてあります

資料	置き場	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
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
用語集	.claude/context/	実働、工数、締め、精算幅などの定義
業務ルール	.claude/context/	締めのロック日、24時間、粗利率の式、パートナーの精算
データ分類とAIへの入力可否	.claude/context/	単価は機密、個人名の扱い
システム構成と接続点	.claude/context/	KINTAIと給与・会計システムの境目
ステークホルダーと承認	.claude/context/	誰が何を決めるか

Copyright © Givery, Inc. All rights reserved

docs/と.claude/context/の使い分け

人が読む正式な資料はdocs/、
AIに常に守らせる要点はcontext/に置きます

観点	docs/	.claude/context/
書く人と読む人	人が書き、人も読む正式な資料	正式な資料から、AIに守らせる要点を抜き出す
長さ	設計書や規約の全文で長い	要点だけで短い
渡し方	作業ごとに @ で指定する	CLAUDE.mdから取り込み、毎回読ませる
例	画面設計書、テーブル定義書、規約	用語集、業務ルール、データ分類

D2-5のSTEP 1では、CLAUDE.mdに.claude/context/の資料への参照を3行足します。

▶ 90秒の動画で見る

Copyright © Givery, Inc. All rights reserved

長い資料を常に読ませない理由

▶ 90秒の動画で見る

CLAUDE.mdから取り込んだ資料も、毎回の依頼に全文が入ります

1	**取り込んでも減らない**	@ で取り込んだファイルも起動時に読み込まれます。整理にはなりますが、窓の使い方は減りません。
2	**200行以内が目安**	CLAUDE.mdは長いほど窓を使い、指示が守られにくくなります。
3	**埋まると守りにくい**	長い資料で窓が埋まると、Claudeは前の指示を忘れたり、誤りが増えたりします。

毎回守らせる要点は.claude/context/に短く書き、長い設計書はdocs/に置いて、作業のときに @ で渡します。

出典　https://code.claude.com/docs/en/memory（2026年10月21日閲覧）
https://code.claude.com/docs/en/best-practices

Copyright © Givery, Inc. All rights reserved

D2-5

コンテキストの見比べ

[25min]

作るもの

CLAUDE.mdへの参照3行、見比べ表（D2-5）

Copyright © Givery, Inc. All rights reserved

ワーク

演習D2-5のねらいと目安

ねらい

AIに渡す資料のない講師の画面と、資料のある自分の画面で、同じ依頼を送ります。
業務ルールをいくつ守れたかを数え、資料の置き場で結果がどう変わるかを確かめます。

できあがりの目安

- [ ] CLAUDE.mdに、.claude/context/の資料への参照が3行ある
- [ ] 見比べ表（D2-5）に、守れたルールの数（4つのうち）が講師側と自分側で書いてある

使うファイルと書くファイル

区分	ファイル	使い方
追記	`CLAUDE.md`	STEP 1で.claude/context/への参照を3行足す
使う	docs/設計/画面設計書_PJ010_収支ダッシュボード.md	作業のときに渡す正式な資料
使う	`.claude/context/`	AIに常に守らせる要点
書く	見比べ/D2-5_コンテキスト.md	守れたルールの数を講師側と自分側で書く

Copyright © Givery, Inc. All rights reserved

ワーク

D2-5の流れ

STEP 1でCLAUDE.mdに参照を3行足してから送ります

■ 自分の手で　■ AIと一緒に　　演習の枠 [25min]

STEP 1 [8min]	STEP 2 [8min]	STEP 3 [6min]	STEP 4 [3min]

■ STEP 1	予想と参照	CLAUDE.mdに参照を3行足して起動し直し、/memoryで3つの資料を確かめます。 予想を1行で書きます。
■ STEP 2	同時に送る	accept edits onにして、講師の合図で同じ文面を送ります。 mvnの承認が出たらEscで断ります。
■ STEP 3	見比べ	講師が4つのルールを示します。書き換えの行を読み、 守れた数を講師側と自分側で数えます。
■ STEP 4	現場での置き換え	自分の現場なら、context/にどの資料の要点を置くかを1行で書きます。

送る文面の全文と参照の書き方は、ハンズオンガイドのD2-5にあります。

Copyright © Givery, Inc. All rights reserved

解説

D2-5で見るところと置き換え

守れたルールの数を、講師側と自分側で数えます

見比べで見るところ（STEP 3）

1. 講師が示す4つのルールのうち、守れた数
2. 講師の画面が推測で埋めたところ
3. 守れたルールが、どの資料のどの行から来たか
4. 自分の画面でも守れなかったルール

自分の現場なら（STEP 4）

毎回守らせたい業務の決まりの要点を、短くして置きます。

- 用語の定義と、計算の式
- AIに入れてよい情報と入れない情報
- 誰が何を決めるかの一覧

守れなかったルールがあれば、足りない資料を考えます。発展のD2-5+で試せます。

Copyright © Givery, Inc. All rights reserved

ハーネスの使い分け

読ませる時点と必ず動かすかどうかで、仕組みを選びます

仕組み	読まれ方と動き方	置き場	向いている用途	今日
CLAUDE.md	常に読まれる	./CLAUDE.md	毎回守る短い決まり	D2-1、D2-5
Skills	条件で読まれる	.claude/skills/	繰り返す手順、参照資料	D2-3
Subagents	別のコンテキストで動く	.claude/agents/	大量に読む作業	D2-2
Hooks	決まった瞬間に必ず動く	.claude/settings.json	止める操作、必ずの検査	D2-4
MCP	外の道具につなぐ	.mcp.json	社内のデータ、外部サービス	デモ

CLAUDE.mdとSkillは依頼で、必ず守らせたい決まりはHookと権限の設定に置きます。

▶ 61秒の動画で見る

Copyright © Givery, Inc. All rights reserved

発展

発展課題

演習	発展課題
D2-2+	spec-reviewerの観点を1つ足し、もう一度レビューさせる
D2-3+	警告メッセージの書き分けを、Skillに足す
D2-4+	止める条件を自分で1つ、Hookに足す
D2-5+	足りないと感じた資料を1枚書いて.claude/context/に置き、もう一度送る

必須のSTEPが終わった方だけ取り組みます。どれもSubagentを並列に動かさない範囲にしてあります。

Copyright © Givery, Inc. All rights reserved

本日のまとめ

> **同じ依頼の結果が、ハーネスと資料で変わることを確かめる**
> **差を生んだファイルを見つけ、自分の現場で書くことを決める**

持ち帰るもの

ファイル	あとで使う場面
自分で育てたCLAUDE.md	自分の現場のプロジェクトで、取り込みと参照の書き方の見本にする
見比べ表5枚	どのハーネスに何を書くかを、チームで決めるときの材料にする
配布フォルダの.claude/	Subagent、Skill、Hookの書き方の見本にする

Copyright © Givery, Inc. All rights reserved

Day3の予告

Day3は、ハーネスを全部そろえた配布フォルダから始めます

演習	呼び名	時間
D3-1	理想的なハーネスの試運転	[20min]
D3-2	成功体験のSkill化	[20min]
D3-3	skill-creatorでのSkill作成	[20min]
D3-4	別テーマでの成果物づくり	[40min]

Day3

11月25日（水）

[180min]

今日育てたCLAUDE.mdと見比べ表は、Day3でハーネスの全体像を見るときに読み返します。

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

[20min]

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/

11月25日（水）のDay3でお会いしましょう。

Copyright © Givery, Inc. All rights reserved