

TRAINING

提案書ができるまでをAIに回す

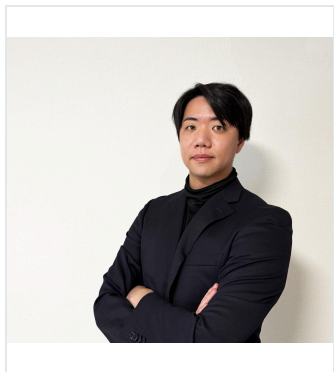
ヒアリングからスライドまで、判断を差し込みながら1周する

アーサー・ディ・リトル・ジャパン株式会社様 2026年8月21日（金） 株式会社
ギブリー 安田 光喜

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。受講者ご本人の学習および社内での検討を目的とした利用に限り、複製と保管を許諾します。第三者への配布、社外での再利用、SNSを含む公開の場への掲載は、事前の書面による承諾なくこれを行わないでください。

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

プロフィール

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。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『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『株式会社シーラクス』にて技術開発責任者として運営。

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
- ・ アプリ開発／自動化プログラムの環境構築
- ・ 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、本日は全力でサポートしていきます。

[#ClaudeCode](#) [#Codex](#) [#Cursor](#)

[#生成AIパスポート](#) [#G検定](#)

本日つくるもの

4つとも、明日の案件でそのまま使えます

ヒアリングツール

HTML ファイル1本。必ず聞く項目をご自身で決めて、それを画面にします。入力すると提案スライドの骨子が出るところで作ります。

提案スライド

ご自身が手で直した骨子から起こした PPTX。テンプレートに沿った形で出ます。

案件フォルダの Skill

次の案件が立ったとき、1コマンドでフォルダ一式と AGENTS.md が揃います。

PowerPoint のアドイン

右側のパネルに、ご自身で足した機能が動きます。

1日の流れ

演習は4本

午前 [150min]

01

振り返り

02

理想フロー

03

シート作成

04

骨子を直す

昼休憩 この間、スライドの生成が動き続けます

午後 [210min]

05

PPTX回収

06

フォルダSkill

07

過去事例

08

24時間

09

議論

10

アドイン

進め方

1 待ち時間は必ず出ます

その間は座学に切り替えます。AI が動いている画面は放っておいて構いません。

2 詰まったら手を挙げてください

同じところで複数名が詰まったら、全体を止めます。

3 実データは入れません

クライアント名、案件名、未公開の数字、個人情報。演習の題材はダミーを用意しています。

3回の位置づけ

貯める・出すと来て、今日は**つなぐ**番です

第1回 貯める



Obsidian の保管庫、調べ物の Skill、CLAUDE.md

第2回 出す



業務をひとつ動くものに。
スライド生成キット

第3回 つなぐ



ヒアリングから提案
スライドまでを1本の流れに

01

第1回・第2回の振り返り

何が手元に残り、何が使われなくなったか

第1回で扱ったもの

Obsidian の保管庫

PMVault。フォルダ構成と、そこへ知識を貯める流れを作りました。

調べ物の Skill

出典つきでノートを作る仕組みを、ご自身で1つ作っていただきました。

CLAUDE.md

毎回書いている前提を書き置いて、指示を短くする仕組みです。

第2回で扱ったもの

業務をひとつ動くものに

/gyomu で、ご自身の業務を形にしました。

スライド生成キット

テンプレートに沿った PPTX が出るところまで。今日も後半で使います。

保管庫を回す

/vault-keep で、貯めたものが古くならないようにする仕組みです。

道具の役割

今日足すのは **Hook** と、案件ごとの **AGENTS.md** の2つです

道具	役割	いつ扱ったか
Claude Code	指示を受けて動く本体	第1回
Obsidian	貯めたものを人が見る窓	第1回
Skill	繰り返す手順を1コマンドにしたもの	第1回・本日の演習②
CLAUDE.md / AGENTS.md	毎回の前提を書き置くファイル	第1回・本日の演習②
Hook	決めた場面で自動的に走る処理	本日

**貯まっているのに、
引き出せていない。**

毎日ニュースが溜まっていくだけで終わっている、という状態はよく起きます。原因は貯め方ではなく、引き出す側の設計にあります。何を聞けば何が返るかが決まっていないと、そもそも開くきっかけが生まれません。

使われるものと、そうでないもの

差を作るのは意志ではなく、**きっかけと手間**の設計です

使い続けているもの

- 開くきっかけが業務の中にある
- 入れる手間より、引き出す得が大きい
- 自分以外にも読む人がいる

開かなくなったもの

- 開くきっかけが無い
- 入れる手間が回収できない
- 作った本人しか中身知らない

今日はここを仕組みで埋めます

Hook で自動的に要約とフォルダを作り、Skill で毎回の手順を1コマンドにします。続けようと意識しなくても続く形にします。

保管庫に何を置くか

貯めるのは資料ではなく、次に使う形にしたものです

生の資料をそのまま置くと、量が増えるだけで引き出せません。

過去の提案書

そのままではなく、要点を書き写した1枚を隣に置きます。全文は検索用、1枚は読ませる用。

うまくいった構成

この業界にはこの順番が通る、という型。案件が終わるたびに1行足します。

使い回せる図表の型

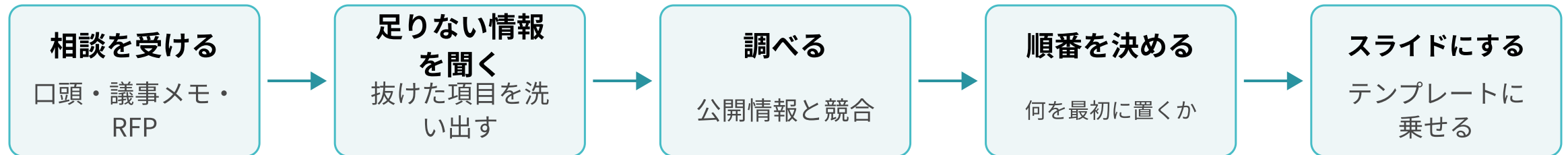
毎回作り直している図があれば、それは型にできます。

聞いておくべきだったこと

提案書を書いていて手が止まった箇所。次のヒアリング項目になります。

相談から提案書までの標準プロセス

今日は2番目から4番目を厚く扱います



スライドを作る速さは、 もう差がつきません。

テンプレートに沿った形なら、骨子さえあれば数分で出ます。差がつくのは、何を聞けたか、何を調べたか、どの順番で言うか。この3つです。ここを AI に丸投げすると、論理は通っているのに「その案件で言うべきこと」からずれたものが出ます。

提案書づくりで、AI を入れても差がつきにくいのはどの工程か。

- A 相手から何を聞き出すか
- B 公開情報をどこまで調べるか
- C 決まった骨子をスライドに落とす
- D どの順番で言うかを決める

正解は次のページで確認します。まず自分で考えてみてください。

正解はC。差がつくのは手前の工程です

C

正解 決まった骨子をスライドに落とす

テンプレートに沿った形なら、骨子さえあれば数分で出ます。ここは誰がやっても同じ結果になります。

A

何を聞き出すか

入力していない情報は AI から出てきません。差が最も大きい工程です

B

どこまで調べるか

どこで足りるかの判断は人が決めます

D

どの順番で言うか

相手が何を気にしているかに依存します。ここも差が出ます

速くなった工程ではなく、速くならない工程に時間を使います

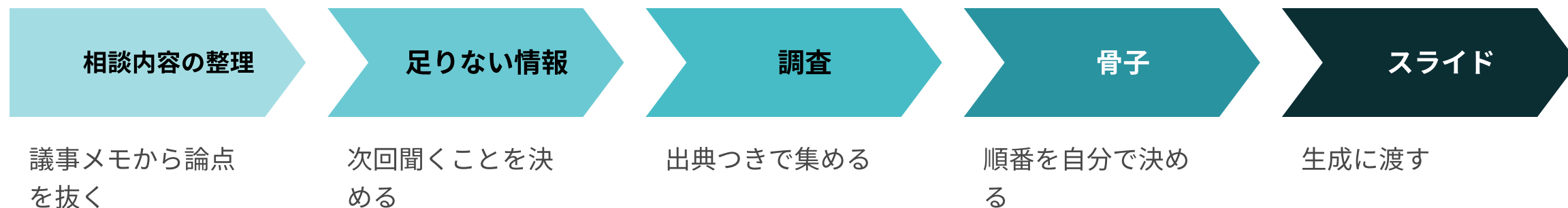
02

提案書作成の理想フロー

どこまで AI に任せ、どこで人が判断するか

段階を分けて指示する

1つの依頼文にまとめて投げない



まとめて投げた場合との違い

まとめて投げたほうも、それらしいものは出ます

1回でまとめて投げる

- それらしいものが出る
- どこで判断がずれたか分からない
- 違ったとき最初からやり直し
- 所要は短い

段階を分ける

- 途中で軌道修正できる
- ずれた段階だけ戻せる
- 各段階の出力が次の入力になる
- 所要は長い

見分け方

その案件でしか言えないことが1つでも入っていれば、段階を分けた甲斐があります。どの会社にも当てはまる文だけなら、まとめて投げたのと同じです。

人が判断を差し込む位置

AI が出すのは案で、決めるのは人です

工程	AI に任せる	人が決める
相談内容の整理	議事メモから論点を抜き出す	どれが本当の論点か
足りない情報	抜けている項目を列挙する	次回どれを優先して聞くか
調査	公開情報を集めて出典をつける	どこまで調べれば足りるか
骨子	順番の案を出す	最初に何を置くか
スライド	テンプレートに沿って生成する	出せる状態かどうか

音声での所感入力

所感は音声で足して、決定は文字に落とします

文字起こしは材料の半分です

録音の文字起こしには、相手の温度感と、席上で言えなかったことが乗りません。そこを打ち合わせに出た人の所感として、声で足します。

文字に落とす境目

順番を決める段階からは文字にします。音声のままだと、前に言ったことを見返せません。

第1回で扱った機能です

操作は覚えていらっしゃると思いますので、今日は使いどころだけお見せします。

音声で投げる時のコツ

最初に前提を1文で言う

「D社という架空の自動車部品メーカーの案件です」から始めると、以降が短く済みます。

言い直しをそのまま残す

整形しようとするとう止まります。「さっきの話ですが」で戻して構いません。

固有名詞は文字で足す

型番や英字の略称は誤変換します。あとから打ち直すほうが速いです。

移動中に投げておく

着いた頃には整理が終わっています。次の項目で扱います。

決めた場面で、 自動的に走る処理。

毎回やっている手作業を、仕組み側へ寄せるためのものです。こちらが指示を出さなくても、決めた場面が来たら勝手に動きます。Skill が「呼ばれたら動く」のに対して、Hook は「その場面が来たら動く」点が違います。

いつ走らせるか

今日は上の2つを実演します。

きっかけ	使いどころ	今日のデモ
やりとりが一段落したとき	その回の要点を残す	議事メモから要約を作る
ファイルを書き出したとき	置き場所を整える	提案書フォルダを用意する
セッションを終えたとき	作業の記録を残す	扱わない
特定のコマンドの前後	確認や後片付け	扱わない

**骨子が固まるまでは、
文字のままにしておく。**

早い段階でスライドの画像まで作ってしまうと、そこから直すのに時間がかかります。画像を見ながら「ここを直して」と指示すると、1ページ直すたびに再生成が走ります。順番が決まるまで文字のままにしておくほうが、結局は速く終わります。

直しのコスト

同じ直しでも、**段階によってコストが変わります**

骨子の段階で直す

- 行を入れ替えるだけ
- 所要は数分
- figma も PowerPoint も開かない

スライドにしてから直す

- 図を作り直す
- レイアウトを調整する
- 1ページごとに再生成が走る

1つの依頼文にまとめて投げず、段階を分けて指示するのはなぜか。

- A まとめて投げると、AI が処理しきれずエラーになるから
- B 出てきたものが違ったとき、ずれた段階だけ戻せるから
- C 段階を分けたほうが、合計の所要時間が短くなるから
- D 1回の指示に含まれる文字数に上限があるから

正解は次のページで確認します。まず自分で考えてみてください。

正解はB。戻る先を作っておきます

B

正解 出てきたものが違ったとき、ずれた段階だけ戻せるから

まとめて投げたほうも、それらしいものは出ます。問題は、どこで判断がずれたかが分からないため、違ったときに最初からやり直しになる点です。

A

処理しきれずエラーになる
長い依頼でもエラーにはなりません

C

合計時間が短くなる
所要は分けたほうが長くなります。やり直しの分を取り返します

D

文字数の上限がある
実務で書く依頼文の長さでは上限に当たりません

その案件でしか言えないことが1つでも入っていれば、分けた甲斐があります

03

演習① 理想的なヒアリングツール

何を必ず聞くかを自分で決め、それを画面にする

提案書を書く手が 止まってから気づく。

初回打ち合わせのあと提案書を書き始めて、手が止まる場所はだいたい決まっています。数字の根拠がない。何をもって成功とするかが決まっていない。予算感が分からない。決裁者が誰か分からない。過去に似た案が落ちていたことを知らない。この5つが空欄のまま帰ると、2回目の打ち合わせが必要になります。

必ず聞く項目（前半）

相手

会社・業界・規模・同席者が何を気にしているか。規模で、比較に出せる他社が変わります。

いまの状態と根拠

「困っている」で止めず、そう判断している数字と時期まで。ここが空欄だと現状分析が1行も書けません。

どうなっていたいか

ありたい姿と、達成したと言える条件。ここを握らないと提案の受け入れ基準が決まりません。

制約

予算・期間・体制・社内の力関係。聞かずに出した案は、だいたい予算か体制で止まります。

必ず聞く項目（後半）

7つは**目安**。自分の領域で必ず聞くものがあれば足す

決める人

目の前の方が決裁者とは限りません。反対しそうな人と、その理由まで。

これまで試したこと

同じ案を出すと即座に落ちます。なぜ止まったかが、こちらの差別化点になります。

次に握ること

次のアクションと期限。この場で決めて帰ります。

増やしすぎない

項目を増やすと打ち合わせ中に埋められません。7つ前後が上限です。「あったら嬉しい」ものは削って、「なければ書けない」ものだけ残します。

根拠を必ず聞く理由

「困っている」で止めない

根拠がないまま持ち帰ると

- 現状分析が書けない
- 他社比較の軸が決まらない
- 提案の規模感が決められない
- 2回目の打ち合わせが要る

根拠まで聞けていれば

- その数字から入れる
- 比較に出す他社が決まる
- 投資規模の目安が立つ
- 次はもう提案の話ができる

聞けていないことを、 AI に埋めさせない。

指示を書かないと、AI はもっともらしい文章で穴を埋めます。埋まってしまうと、聞けていないことに気づけません。次の打ち合わせで同じ質問をする羽目になります。空欄は【要確認】と書いて残す。これを作りの側で決めておきます。

「ご提案」も自動で出さない

入力した事実からは提案は出てきません。出てきたとしたら、それは一般論です。ここは人が書く欄として空けておきます。

演習① ヒアリングシートを作る

目的 何を必ず聞くかを自分で決めて、それを画面にする

[50min]

やること

- 1 直近の案件を思い出して、必ず聞く項目を決める [10min]
- 2 要件メモ.md に自分の手で書く [10min]
- 3 Claude Code に渡して作らせる [25min]
- 4 実際に入力して、欄と順番を直す [5min]

できあがるもの

hearing.html (HTML ファイル1本)

ここまで来たら次へ

決めた項目がすべて画面にある。入力した内容がページを閉じてても残る。送信ボタンを押さないと消える作りになっていない。

4つ目を書かないと、フォームの慣習でボタンが付きます

単一のHTMLファイル1本にまとめる

ファイルが分かれると、渡すときと開くときに手間が増えます。

外部ライブラリとCDNは使わない

会社の回線や端末の設定で読み込めないことがあります。

入力内容はブラウザに保存する

打ち合わせ中に使うものなので、閉じても残る必要があります。

送信ボタンは作らない

押し忘れると消えます。入力した瞬間に残る作りにします。

骨子の型

今日は結論を先に置く形で出します

どれが正解かではなく、相手が何を先に知りたいかで選びます。

型	並べ方	向く場面
SCQA	状況 → 複雑さ → 問い → 答え	相手がまだ問題を認識していないとき
ピラミッド	結論 → 根拠3つ → 各根拠の裏づけ	結論を先に聞きたい相手
時系列	現在 → 短期 → 中期 → 長期	ロードマップを示すとき
対比	現状 → あるべき姿 → 差分 → 埋め方	変革の必要性を出すとき

骨子を出して、手で直す

そのまま使えるものは、**まず出ません**

1 入力から骨子が出るようにする

何を根拠に何を並べるかまで指定します。「入力欄の内容をもとに」だけだと一般論が出ます。

2 ダミー案件を入れる

架空の産業機械メーカーの案件を用意しています。実データは使いません。

3 出てきた箇条書きを手で直す

順番を入れ替える。言い過ぎを削る。抜けている論点を足す。ここが今日の山場です。

直すときに見るところ

直した箇所が、次にシートへ足す項目になります

- 最初に置くべき結論が、最初に来ているか

AI は時系列や網羅の順で並べます。相手が最初に知りたいことは別です。

- 相手が一番気にしていることが、途中で埋もれていないか

入力していない情報なので、AI には重みが分かりません。

- 根拠のない主張が混ざっていないか

出典のない一文は、そのページごと落とします。

- この案件で言う必要のない一般論が入っていないか

どの会社にも当てはまる文は、読み手の時間を奪うだけです。

スライド生成に渡す

承認を返してから昼休憩へ

- 1 **/adl_deck2 と打ってから、直した箇条書きを貼る**
整形し直す必要はありません。末尾に「書いていないことは足さないでください」の1行を足します。
- 2 **ページの並びが出たら、確認して承認を返す**
キットは生成の前に並びを見せて止まります。返事をしないと、昼の間なにも進みません。
- 3 **承認を返してから、席を立つ**
PC は閉じず、スリープさせないでください。生成は休憩の間に走ります。

「書いていないことは足さない」が効きます

これがないと、体裁を整えるために一般論のページが挟まります。あとから消すほうが手間です。

**90%合っていれば良い、
という基準では通らない。**

1件あたりの金額が大きい仕事では、資料の体裁そのものが品質の一部として見られます。90%合っているという状態は、10%外れている状態です。文字の大きさ、余白、ロゴの位置、出所の書き方。1つでもずれていれば、そのまま社外には出せません。

直すか、作り直すか

この判断を**1回**やっておくと、次から迷いません

手で直したほうが速い

- ずれが3か所以内
- 文字の大きさや余白のずれ
- 1ページだけの問題
- 骨子そのものは合っている

作り直したほうが速い

- ずれが4か所以上
- ページの順番が違う
- 言っていないことが足されている
- 骨子から見直す必要がある

ヒアリングシートで、入力されていない項目の扱いとして正しいのはどれか。

A AI に推測させて埋め、あとから人が確認する

B 空欄のまま何も表示しない

C 【要確認】と表示して残す

D その項目ごと骨子から削除する

正解は次のページで確認します。まず自分で考えてみてください。

正解はC。埋めると気づけなくなります

C

正解 【要確認】と表示して残す

指示を書かないと、AI がもっともらしい文章で穴を埋めます。埋まると気づけません。

A

推測させて、あとから確認

確認する前提でも、埋まった文章は見過ごされます

B

空欄のまま何も出さない

抜けていること自体が見えません

D

項目ごと削除する

次回聞くべきことが分からなくなります

何が聞けていないかが、次のヒアリングの準備になります

04

演習② 案件フォルダを用意する Skill

毎回手で作っているフォルダを、1コマンドにする

日本語で書いた 手順書が1枚あるだけ。

SKILL.md というファイル1枚で、Claude Code に「この名前と呼ばれたらこう動く」を教えています。置き場所は ~/.claude/skills/ の下。ここに置くと、スラッシュを打ったときに候補に出ます。プログラムを書く必要はありません。

SKILL.md の中身

description の書き方で、呼べるかが決まります

場所	書くもの	効き方
先頭の name	呼び出す名前	/名前 で呼べるようになる
先頭の description	どんなときに使うかを日本語で	ここに書いた言葉で発火する
本文	やること、守ること、注意点	呼ばれたときに読まれる
同じフォルダの補助ファイル	テンプレート、参照資料	必要に応じて開かれる

description の書き方

自分が**実際に打つ言葉**を、そのまま入れます

呼べなくなる書き方

- 「便利なツール」
- 「作業を効率化する」
- 抽象的すぎて発火しない
- 自分が打つ言葉が入っていない

呼べる書き方

- 「案件フォルダを作って」
- 「新しい案件」
- 実際に打つ言葉をそのまま入れる
- 具体的すぎても呼べなくなる

似た仕組みの違い

違いは、**呼ぶか勝手に走るか**です

仕組み	いつ動くか	使いどころ
Skill	名前と呼ばれたとき	手順が決まっている作業を1コマンドに
Hook	決めた場面が来たとき	毎回やっている後片付けや記録
CLAUDE.md	常に読まれる	全案件に共通する前提
AGENTS.md	そのフォルダで作業するとき	案件ごとの前提

案件フォルダのひな形

これが**正解**というわけではありません

フォルダ	入れるもの	いつ触るか
01raw	もらった生データ。議事メモ・RFP・先方提供資料	案件が立った直後
02research	調べたもの。公開情報・競合・市場データ	調査フェーズ
03plan	ストーリーラインとスライドプラン	骨子を固めるとき
04assets	図表・画像	スライドを組むとき
05pptx	出力	納品前
AGENTS.md	この案件でのAIの動き方	最初に1回、あとは随時

他の分け方

自分の実務でどこを変えるかを考えてください

クライアント別

同じ相手で案件が続く場合。過去のやりとりが1か所にまとまります。

年度別

実績を年度で振り返る必要がある場合。ただし進行中の案件が探しにくくなります。

フェーズ別

提案前と受注後で分ける形。関わる人が変わるなら有効です。

番号を振らない

01rawのような番号は並び順のためです。検索で開くなら不要になります。

AGENTS.md に何を書くか

ここがこの演習の本体です

出所の付け方

数字には出典と公表日を必ず添える。出所が確認できない数字は書かない。

使ってよい情報の範囲

このフォルダの中だけか、公開情報も含めるか。扱わないものも明記します。

避けたい表現

断定を避ける箇所、売り文句を書かない、見出しは体言止め。案件ごとに変わります。

参照させたいもの

過去の類似案件、社内テンプレート。パスで指定します。

フォルダを作るだけなら手でもできます

効くのは、その案件で AI に守らせたいことを書き置ける点です。案件が変わればここだけ書き換えます。

演習② Skill を作る

目的 新しい案件が立ったときに、1コマンドでフォルダー式が整う状態にする

[40min]

やること

- 1 SKILL.md の中身を実物で見る [5min]
- 2 設計メモ.md にフォルダ構成と AGENTS.md の中身を決める [10min]
- 3 Claude Code に渡して Skill を作らせる [20min]
- 4 空のフォルダで呼んで、想定どおりか確かめる [5min]

できあがるもの

~/claude/skills/ に置いた Skill

ここまで来たら次へ

コマンドを1つ打つとフォルダー式ができる。
AGENTS.md に中身がある。別の案件名でもう一度実行できる。

案件フォルダの Skill で、AGENTS.md に書く内容として適切なのはどれか。

- A フォルダの一覧と、それぞれの名前の由来
- B 案件の進捗状況と、次回打ち合わせの日程
- C 使用しているソフトウェアのバージョン一覧
- D その案件で守らせたいこと。出所の付け方、使ってよい情報の範囲、避けたい表現

正解は次のページで確認します。まず自分で考えてみてください。

正解はD。書き換えるのはここだけです

D

正解 その案件で守らせたいこと。出所の付け方、使ってよい情報の範囲、避けたい表現

フォルダを作るだけなら手でもできます。効くのは、AI に守らせたいことを書き置ける点です。

A

フォルダの一覧と名前の由来
書いても AI の動きは変わりません

B

進捗状況と次回の日程
変化が速すぎて、書いた翌週には古くなります

C

ソフトウェアのバージョン一覧
この Skill の目的と関係がありません

AGENTS.md を空で作らせないことを、指示に必ず入れてください

05

演習③ 過去事例の読ませ方

同じ問いを4通りで投げる。読めていない資料の見つけ方まで

4つの読ませ方

速さと漏れの少なさは、たいてい逆を向きます

方法	やること	得意	苦手
Wiki 型	人が要約して1枚にまとめる	速い・正確	まとめた時点で止まる
RAG 型	全文を検索させる	更新が楽	聞き方で拾えない
Agentic RAG 型	AI がフォルダを歩いて自分で開く	漏れにくい	遅い
画面の保存	ページを画像で撮って渡す	図表が読める	引用の精度が落ちる

聞かれてから、
関係する箇所を**探して答える**。

資料全部を覚えているわけではありません。質問を受けてから、似ている箇所を検索して、見つかった部分だけを読んで答えます。だから資料を足しても学習し直す必要がなく、更新が楽です。反面、検索で引っかからなかった箇所は、存在しないのと同じ扱いになります。

RAG が拾えない場面

4つ目が今日の本題です

言い回しが違う

資料に「調達コスト」と書いてあり、質問が「仕入れ値」だと引っかけからないことがあります。

複数の資料にまたがる

1本ずつには書いていないが、3本並べて初めて見える論点は拾えません。

図表の中にしかない

本文になく、図の中だけに書かれた数字は検索に出てきません。

そもそもテキストが無い

画像だけの資料は、検索対象になりません。今日はここを扱います。

RAG と Agentic RAG

違いは、**探す主体**が人か AI かです

RAG

- こちらが検索条件を決める
- 引っかけた箇所だけ読む
- 速い
- 聞き方に結果が左右される

Agentic RAG

- AI が何を見るかを決める
- 必要と判断したファイルを順に開く
- 遅い
- 漏れにくい、開いた分だけ時間がかかる

まとめた時点で、 更新が止まる。

人が要約して1枚にまとめる方法は、速くて正確です。2回目以降は数秒で終わります。ただし、まとめたあとに足された資料は入っていません。半年前に作ったサマリを読ませて、最新の案件が反映されていないことに気づかない。これが起きます。

使う素材

1本だけ、AI から見ると中身が空っぽです

3本とも経済産業省向けの調査報告書です。見た目はどれも同じ PDF で、開けば読めます。

報告書	ページ	文字が取れるページ	1ページあたり
2025 工作機械のルール形成戦略	92	92 / 92	508字
2024 蓄電池のデューデリジェンス	78	78 / 78	581字
2023 バイオ医薬品と再生医療の技術動向	187	1 / 187	1.1字

PDF の構造

1280×720 の**写真が1枚**、貼ってあるだけです

テキスト層

- 文字がそのまま入っている
- 検索・コピーができる
- AI が読むのはここ
- 2025年と2024年の2本にはある

画像層

- 背景・写真・グラフの絵
- 見た目を作っている部分
- 中の文字は読めない
- 2023年のものはページ全体がこれ

フォントの有無では判定できません

2023年のものにも MS-Mincho が埋め込まれています。最終ページの奥付だけがテキストで、そこで使われていました。

読めていないことに、
気づけない。

AIは「このファイルは読めませんでした」とは言いません。他の2本の情報だけで答えを組み立てて、それらしい回答を返します。答えが返ってくるので、1本まるごと落ちていることに気づけません。

該当しやすい資料

紙をスキャンしたもの、印刷用書き出したもの、他社から受け取った資料、セキュリティのために画像化したもの。

判定のしかた

数字で判定する

1 Claude Code に聞く

「このフォルダの PDF を1本ずつ調べて、テキストが取り出せるものと画像だけのものに分けてください」

2 1ページあたりの文字数を見る

10字を切っていたら、まず画像です。まともな報告書なら500字前後になります。

3 フォントの有無では判断しない

画像PDFにもフォントが埋め込まれていることがあります。

画像にして読ませる

さっきまで1文字も読めなかったものが、**読めます**

1 必要なページを画像にする

画面を撮るか、Claude Code に書き出させます。解像度は110～150dpi。

2 渡すのは20ページまで

187ページを丸ごと渡すとうまくいきません。目次で当たりをつけて、使う章だけ渡します。

3 図表の数値まで読める

円グラフのパーセンテージも、棒グラフの上の数値も、脚注の出所も返ってきます。

解像度の目安

上げても読める内容は**変わりません**

設定	1ページの大きさ	読めるもの	使いどころ
96dpi	1280×720	本文と見出し	文章中心の資料
110dpi	1467×825	脚注と図表の数値	今日の推奨
150dpi	2000×1125	同上	細かい表がある場合
300dpi	4000×2250	同上（変わらない）	印刷用。AI に渡す意味はない

OCR という選択肢

読み取った数字は、**必ず元のページで確かめます**

向く場面

- 同じ資料を何度も参照する
- 全文検索できるようにしたい
- テキストとして引用する必要がある

注意すること

- 表と図が多いと読み違いが混ざる
- 数字を1桁違えると結論が変わる
- OCR後は必ず元ページと突き合わせる
- 187ページ全部にかけるのは重い

中身は **ZIP と XML**。
画像認識ではない。

拡張子を .zip に変えると展開できます。スライド1枚が1つの XML で、画面の文字はタグに挟まれてそのまま入っています。AI はここを読んでいます。だから PDF より速く、正確に読めます。

PPTX から読めるもの

中身	読めるか	理由
テキストボックスの文字	読める	XML に文字として入っている
図形の中の文字・表のセル	読める	同上
スピーカーノート	読める	別の XML に入っている
貼り付けた画像の中の文字	読めない	写真として格納されるだけ
他ソフトのグラフを画像で貼ったもの	読めない	同上

左右の対比も、 矢印の向きも**伝わらない**。

XMLに入っているのは、文字と、それを置いた座標です。どれとどれが並んでいるか、何を対比しているかは書かれていません。だから要約させると、書いてある文字は拾えるのに、その資料が何を主張しているかがぼやけます。

対策は2つ

見出しに主張を文章で書いておく。図に語らせない。もうひとつは、要約させるときにページの画像もあわせて渡すこと。文字はXMLから、構造は画像から読ませます。

保管するならどの形か

PPTX と md を並べて置くのが実用的

全体像は md、細部は PPTX。AI にはまず md を読ませ、必要なら PPTX に降りさせます。

形	AI から見て	使いどころ
PPTX のまま	文字は完全に読める。構造は弱い	元ファイルがあるなら、これが最良
PDF（テキストあり）	読める。ページ番号も取れる	配布用。引用にも使える
PDF（画像のみ）	読めない	避ける。やむを得なければ画像で渡す
要点を md に書き写す	完璧に読める。人の手が要る	繰り返し参照するものだけ

保管庫のどこに置くか

置き場所は、次に誰が引くかで決まります

■ 案件フォルダの中に置く

その案件でしか使わないものは、案件フォルダの 02research に。

■ 横断で使うものは保管庫へ

業界の構造、繰り返し使う図表の型、うまくいった構成。

■ 要点の md を必ず添える

PDF だけを置くと、探すときに開いて確かめることになります。1枚あれば一覧で判断できます。

■ 入れないもの

顧客の非公開情報。ポリシー上まだ扱えないので、公開版と架空のダミーで組みます。

演習③ 4通りで投げて比べる

目的 同じ問いを読ませ方だけ変えて投げ、答えの差を見る

[60min]

やること

- 1 素材を見る／4つの読ませ方を知る [10min]
- 2 同じ問いを4通りで投げる [25min]
- 3 なぜ1本だけ読めなかったのかを知る [5min]
- 4 画像にして読ませ直す [10min]
- 5 PowerPoint の場合を知る [5min]
- 6 手元の資料に同じことが起きていないか考える [5min]

できあがるもの

4通りの出力の比較（比較シート.md）と、手元の資料の判定

ここまで来たら次へ

同じ問いを4通りで投げた。読めなかった1本がどれか分かった。画像として渡し直したら読めることを確かめた。

元の PowerPoint を 探すのが正解。

OCR にかけると、表と数字の読み違いを全ページ確認することになります。画像で渡すなら、必要なページを選ぶ手間が毎回かかります。この報告書は今後も参照されるので、1回だけの手間で済ませるべきです。

見つからない場合の順番

目次と扉ページだけ画像で読ませて構造を把握 → 使いたい章だけ OCR にかけて数字を目視確認 → その章の要点を md に書き写して、以後はそちらを使う。

過去案件3件から共通する論点を漏れなく拾いたい。最も向いている読ませ方はどれか。

- A 人が要約した1枚のまとめを読ませる
- B AI にフォルダを歩かせ、必要なファイルを自分で開かせる
- C 検索で引っかかった箇所だけを読ませる
- D 3件のスライドを画像にして読ませる

正解は次のページで確認します。まず自分で考えてみてください。

正解はB。漏れの少なさなら Agentic RAG です

B

正解 AI にフォルダを歩かせ、必要なファイルを自分で開かせる

漏れにくさを優先するなら Agentic RAG です。時間はかかります。速さを取るか漏れの少なさを取るかで選び分けます。

A

人が要約した1枚

速くて正確ですが、まとめを作った時点の内容しか入っていません

C

検索で引っかかった箇所

聞き方に依存し、言い回しが違うと拾えません

D

画像にして読ませる

図表には効きますが、本文の細部は落ちます

急ぎの打ち合わせ前なら Wiki 型、提案の骨子を作るなら Agentic RAG です

06

AI 時代の24時間

自分が画面を見ていない時間に、何が進んでいるか

使える時間

重い処理は、自分が触っていない時間に寄せます

移動中

スマホから音声で投げておきます。着いた頃には調べ物が終わっています。

会議の合間

次の会議までの15分に、長い処理を投げます。

就寝中

1～2時間かかるものは夜に回します。朝には出ています。

AI が全力で動くと、 PC が重くなる。

自分が別の作業をしている間に走らせると、どちらも遅くなります。だから「自分が触っていない時間にしかできないこと」を選んで投げるほうが、結果として1日に進む量が増えます。

スマホから投げる時のコツ

結果の受け取り方を決めておく

朝いちばんに何を見れば済むかを、投げる前に指定します。

途中で聞かれない形にする

「不明な点は仮置きして最後にまとめて質問してください」と足します。

題材は前日に決めておく

移動中に題材から考え始めると、投げる前に着きます。

目的 優秀な部下に、寝ている間なにをやらせるか

[25min]

やること

- 1 その部下は優秀で、時間の制約がなく、こちらが寝ている間も動き続ける
- 2 ただし途中で質問はできない。朝、結果だけが出ている
- 3 この条件で、何を任せるか

できあがるもの

自部門で夜間に回せそうな仕事のリスト

ここまで来たら次へ

判断が要らない仕事から挙がっているか。手順が決まっていて量が多いものが入っているか。

論点として置くもの

判断が要らない仕事から出す

■ スライドの作成

1〜2時間かかるもの。骨子まで決めておけば、朝には形になっています。

■ 大規模な調査

翌日の会議に向けた事前の調べ物。予定を読んで、当日分を先に調べておく形もあります。

■ 既存システムの改修や整理

動作確認まで含めて。人が見るのは結果の差分だけです。

■ 上の3つを回す仕組みそのもの

毎晩同じことをやらせるなら、その手順を Skill にしておきます。

07

演習④ PowerPoint のアドイン

ゼロからは作らない。動くものを起動して、機能を1つ足す

HTML と JavaScript。 特別なことはしていない。

PowerPoint の右側に出るパネルは HTML でできていて、動きは JavaScript です。貼り付けた骨子を PowerPoint への命令（Office.js）に変換して、その場でページを足しています。外部とは通信しません。今日は完成品を起動するところから始めて、機能を1つだけ足します。

アドインの中身

中身は演習フォルダで開いて確かめられます。

ファイル	書いてあること
manifest.xml	どこに表示するか、何の権限が要るか
taskpane.html	パネルの見た目
taskpane.js	ボタンを押したときの動き
package.json	起動コマンド

演習④ 機能を1つ足す

目的 動くアドインを起点に、自分で機能を追加する

[35min]

やること

- 1 配布フォルダの完成品を起動する [10min]
- 2 パネルからプロンプトを送って、スライドが出るのを見る [5min]
- 3 テンプレプロンプトから機能を1つ足す [15min]
- 4 社内に配るときに何が要るかを聞く [5min]

できあがるもの

機能を1つ足したアドイン

ここまで来たら次へ

完成品を起動してパネルが出た。パネルからスライドを生成した。機能を1つ足した。

社内に配るときに要るもの

ハードルは、作るのではなく**配る**ことです

手元で動かす	社内に配る	なぜ
Node.js	同じ	動かす土台
ローカルの証明書	正式な HTTPS 証明書	Office は HTTPS でないと読み込まない
localhost で配る	全員から見える場所に置く	localhost は自分の PC の中だけ
自分で登録	管理センターから配布	組織配布には管理者権限が要る

3ヶ月使い続けたものだけを、 情報システム部門へ。

いきなり全社に配ろうとすると、承認だけで数ヶ月かかります。まず自分と、隣の席の1人で使ってみる。続かなければ配る必要がありません。続いたものだけを持っていくほうが、話が速く進みます。



08

持ち帰り

明日からの1本目をどうするか

手元に残ったもの

■ ヒアリングツール

次の案件の初回打ち合わせで、その場で開いて埋められます。

■ 提案スライド

自分で直した骨子から起こした PPTX。作り方の一連が手元にあります。

■ 案件フォルダの Skill

次の案件が立ったとき、1コマンドで整います。

■ PowerPoint のアドイン

作れる範囲の感覚が手に入りました。

言えるようになったこと

どこを AI に任せ、どこで自分が判断するか

相談内容の整理と情報の収集は任せ、言う順番の決定は自分でやります。

ヒアリングで何を必ず聞くか

7項目を自分の言葉で決め、それを画面にしました。

過去の資料をどの形で置いておくか

速さを取るか漏れの少なさを取るかで、置き方が変わります。

自分が触っていない時間に何を任せるか

判断の可否で分けます。

次の案件で、
ヒアリングツールを**1回使う**。

持ち帰りはこの一点です。使ってみると足りない欄が必ず出ます。そこを足すのが2回目で、そこから先は毎回速くなります。作り直す必要はありません。

語	意味
Skill	呼ばれたときに動く手順書。SKILL.md というファイル1枚
Hook	決めた場面が来たら自動的に走る処理
CLAUDE.md	全案件に共通する前提を書き置くファイル
AGENTS.md	その案件だけの前提を書き置くファイル
プランモード	作業を始める前に、何をするかの計画だけを出させる使い方
ストーリーライン	スライドにする前の、言う順番を決めた箇条書き

語	意味
RAG	質問を受けてから資料を検索し、見つかった箇所だけを読んで答える仕組み
Agentic RAG	AI 自身がどのファイルを開くかを決めて読みにいく形
OCR	画像の中の文字を読み取ってテキストにする処理
Office.js	PowerPoint や Excel の中で動くアドインを作るための仕組み
タスクペイン	Office の右側に出る作業パネル
localStorage	入力内容をブラウザの中だけに保存する仕組み。外部には出ない



株式会社ギブリー (Givery, Inc.)

〒150-0036 東京都渋谷区南平台町15-13 帝都渋谷ビル8F