

ハンズオンガイド

Day1 ヒアリングメモから要件定義書へ

改修依頼書と3本のヒアリングメモから、根拠の行番号が付いた要件一覧を作り、Word の要件定義書にします。どの演習も、最初の数分は AI を開かずに自分で書きます。

2026年10月8日（木）

Zoom

個人演習

Claude デスクトップアプリ

目次

本日のゴール

3回の見取り図

当日の流れ

Zoom での進め方

題材の歩き方

サンプル情報サービス株式会社（架空）

改修依頼の4項目

登場人物

現行画面

配布フォルダの構成

手書きメモの置き場所

架空データの扱い

Code タブの使い方

フォルダの開き方

@ によるファイル指定

モードの切り替え

差分の見方

コマンド実行の許可

D1-1 文脈の有無による出力の違い

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 渡すものと禁じることのメモ**STEP 2** 3通りの依頼**STEP 3** メモにない要件と根拠の確かめ**STEP 4** 違いの3点と指示の型

うまくいかないとき

早く終わった方へ

D1-2 ヒアリングメモからの要件抽出

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 要件候補の手書き**STEP 2** Claude による抽出**STEP 3** 自分の候補との突き合わせ**STEP 4** 要件一覧の確定

うまくいかないとき

早く終わった方へ

D1-3 要件定義書のドラフト作成

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 書きにくい章の見当**STEP 2** Markdown での下書き**STEP 3** Word への変換と確認**STEP 4** 要確認の数え直し

うまくいかないとき

早く終わった方へ

D1-4 抜け漏れ探し

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 勤怠 CSV からの抜け探し**STEP 2** Claude による抜け漏れ探し**STEP 3** 自分の発見との比較**STEP 4** v0.2 と確認事項の作成

うまくいかないとき

早く終わった方へ

よくある質問

01

本日のゴール

3本のヒアリングメモと改修依頼書から、根拠の行番号が付いた要件一覧と、Word の要件定義書 v0.2 を作って帰ります。

持ち帰るもの

- ☐ 成果物/day1/要件一覧.md 。要件が25件以上あり、全件にファイル名と行番号の根拠が付いている
- ☐ 成果物/day1/要件定義書_v0.2.docx と、その元の 要件定義書_v0.2.md 。書けない箇所は空けずに「要確認」と確認相手を書いてある
- ☐ 成果物/day1/確認事項.md 。誰に、何を、なぜ聞くかが5件以上
- ☐ 成果物/day1/D1-1_比較.md の末尾に、自分の言葉で書いた指示の型が1行

Day2はこの要件定義書を入力にして画面を設計します。v0.2まで届かなかった方には、Day2の最初に講師が合流版を配ります。自分の成果物が途中で、そこまでのファイルは消さずに残してください。

3回の見取り図

回	入力	作るもの
Day1 要件定義	ヒアリングメモ3本、改修依頼書、勤怠工数CSV	要件一覧、要件定義書 v0.2、確認事項
Day2 画面設計	Day1の要件定義書、現行画面、経理のヒアリングメモ	画面一覧と遷移、KIN030の画面設計書、モック、請求回収の業務フロー
Day3 レビューと実装	Day2の設計書、別チームのPJ010設計書、勤怠と原価のデータ	レビュー記録、対応表、収支ダッシュボード、レビュー会資料、利用ルール

3回とも、同じ架空の会社の同じシステムを扱います。前の回で作ったものが次の回の入力になるので、ファイル名は手順のとおりにつけてください。

02

当日の流れ

全体で [180min] です。講義のあとには、必ず演習か理解度チェックが入ります。

枠	中身	形式	時間
導入	3回の見取り図、題材（KINTAIと改修依頼）、現行画面に触る、Codeタブで配布フォルダを開く、架空データの注意	説明	[15min]
D1-1	文脈の有無による出力の違い	個人演習	[25min]
D1-2	ヒアリングメモからの要件抽出	個人演習	[40min]
休憩			[10min]
D1-3	要件定義書のドラフト作成	個人演習	[35min]
D1-4	抜け漏れ探し	個人演習	[25min]
まとめ	指示の型3つ、要件定義書のどこを疑うか、理解度チェック	まとめ	[15min]
質疑・アンケート		質疑	[15min]

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。手元では Claude デスクトップアプリと、エクスプローラー、メモ帳、LibreOffice を使います。画面が1枚の方は、Zoom の画面を小さくして横に並べると作業しやすくなります。
- 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 変換で文字化け」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。画面を見る必要があるときは、講師から画面共有をお願いすることがあります。
- 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。途中でも、次の STEP へ移って構いません。終わらなかった分は、研修のあとに同じ手順で続けられます。

題材の歩き方

3回とも、同じ架空の会社の、同じシステムの改修を扱います。導入の時間に、ここを一緒に見ます。

サンプル情報サービス株式会社（架空）

独立系の Sler で、社員は約300名です。第1開発部が金融、第2開発部が公共と運輸の案件を持ち、ほかに管理部があります。

この会社で全社員が使う唯一の Web システムが、勤怠・工数管理システム KINTAI です。2014年に稼働し、JSP、Spring MVC、MyBatis で作られています。TERASOLUNA 5.x がベースです。研修で Java のコードを読む場面はありません。TERASOLUNA は Day2 で、入力チェックの書き分け方の決まりとしてだけ使います。

改修依頼の4項目

情報システム課が起票した `資料/day1/改修依頼書_KINTAI-2026-03.md` に、次の4項目が書いてあります。

- 1 工数入力画面（KIN030）に「予定工数」を入力できるようにする
- 2 1日の作業時間の合計が24時間を超えたら入力エラーにする
- 3 プロジェクトの収支を見る画面（PJ010 収支ダッシュボード）を新設する
- 4 希望時期は2027年4月。詳細は各部のヒアリングで確認すること

4つ目にあるとおり、依頼書だけでは要件になりません。誰が、いつ、何に困っていて、どこまで直せばよいのかは、ヒアリングメモの中にあります。しかも、メモ同士で言っていることが合わない箇所があります。

登場人物

人物	所属	出てくる資料	受け持ち
田中課長	第1開発部 課長	<code>ヒアリングメモ_01_第1開発部_田中課長.md</code>	金融の2案件、PJ-2026-011 と PJ-2026-014 の PM
鈴木主任	第2開発部 主任	<code>ヒアリングメモ_02_第2開発部_鈴木主任.md</code>	電力の2案件、PJ-2026-017 と PJ-2026-071 の PM
佐藤さん	管理部 担当	<code>ヒアリングメモ_03_管理部_佐藤さん.md</code>	管理部で勤怠と原価のデータを扱う
山本さん	管理部経理 担当	<code>ヒアリングメモ_04_管理部経理_山本さん_請求回収.md</code>	請求と入金を扱う。Day2 の D2-4 で登場

人物は名字と役職だけで表します。取引先は「A社（架空・信託銀行）」の形で、実在の会社ではありません。

現行画面

配布フォルダの **現行画面/index.html** をダブルクリックすると、ブラウザで KINTAI の現行画面のモックが開きます。サーバは要りません。導入の時間に、次の5画面を一通り触ってください。

画面ID	画面名	使う人	中身
KIN010	メニュー	全員	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	全員	勤務区分、出勤、退勤、休憩（分）、実働（自動計算）
KIN030	工数入力	全員	日付ごとに、プロジェクト（選択）、作業区分、作業時間（0.25時間単位）、備考を行で入れる。合計は表示するが、チェックはしない
KIN040	交通費申請	全員	日付、区間、金額、目的
KIN050	月次締め・CSV	PM・管理部	締め状態、CSV 出力、パートナー実績の CSV 取込



KIN010 メニュー



KIN020 日次勤怠入力



KIN030 工数入力



KIN040 交通費申請



KIN050 月次締め・CSV

触るときに見ておくこと

- ☐ KIN030 で **行を追加** を押し、プロジェクトと作業区分を選んで作業時間に「25」を入れ、**保存** を押してみる。どのエラーが、画面のどこに出るか
- ☐ KIN020 の実働が8時間の日に、KIN030 で合計10時間の工数を入れてみる。エラーになるか
- ☐ KIN050 で、締め状態とロック予定の日時がどう書いてあるか

ここで見たことは、D1-2 以降で要件を読むときの手がかりになります。気づいたことはメモしておいてください。

配布フォルダの構成

事前準備で、配布 ZIP をデスクトップに展開してあります。Claude に渡すファイルも、作ったファイルの置き場所も、すべてこのフォルダの中です。

```
idup-handson/
├── はじめにお読みください.md
├── CLAUDE.md      Claude が最初に読む。AGENTS.md を読み込む
├── AGENTS.md      この研修での約束ごと
├── 資料/          読むだけ。書き換えない
│   ├── day1/      ヒアリングメモ3本、改修依頼書
│   ├── day2/      経理のヒアリングメモ
│   ├── day3/      レビュー対象の画面設計書
│   ├── データ/     勤怠工数CSV、マスタ、売上実績表、請求回収
│   └── 参考/       TERASOLUNA の書き分け、用語集
├── 現行画面/      KINTAI の現行画面。ブラウザで開く
├── テンプレート/  要件定義書、画面設計書などの雛形6本
├── tools/         Markdown を Word・Excel・PowerPoint にする変換ツール
├── 成果物/
│   ├── day1/ day2/ day3/ 作ったものはここに置く
│   └── 合流用/      講師が配る合流版を入れる場所
```

場所	扱い
資料/	読むだけです。Claude にも書き換えさせません。AGENTS.md にそう書いてあります
テンプレート/	要件定義書、画面設計書、レビュー記録、対応表、レビュー会資料、利用ルール雛形です。コピーして使うのではなく、Claude に渡して形をそろえさせます
tools/	Claude が Markdown を Word (docx)、Excel (xlsx)、PowerPoint (pptx) に変換するときに使います。自分で実行する必要はありません
成果物/day1/ など	自分で書いたメモと、Claude に作らせたファイルを置きます。研修が終わってもそのまま持ち帰れます

Claude はフォルダを開くと **CLAUDE.md** を読み、そこから **AGENTS.md** の約束ごとを読み込みます。回答は日本語です・まず調、ヒアリングメモにないことを要件として足さない、推測で埋めた箇所には「(要確認)」を付ける、要件や指摘には **ヒアリングメモ_01:L23** の形で根拠を付ける、の4つが要です。送る文面の中で毎回くり返さなくても、Claude はこれを守ろうとします。

手書きメモの置き場所

どの演習も、STEP 1 は AI を開かずに自分で書きます。書く道具は Windows のメモ帳です。書いたメモは、STEP 2 で Claude に送る文面の **<ここに…>** のところへ貼ります。

- スタートメニューで「メモ帳」と入れて開きます。
- メモを書いたら、**ファイル** → **名前を付けて保存** を選びます。
- 保存先に **idup-handson** の中の **成果物** → **day1** を選びます。

4 ファイル名を **メモ_D1-1.md** のように「メモ_演習番号.md」にします。**ファイルの種類** は **すべてのファイル (*.*)**、**エンコード** は **UTF-8** を選んで保存します。

Tips D1-2 の STEP 1 だけは、ファイル名が **自分の要件候補.md** に決まっています。ほかの演習は「メモ_演習番号.md」で揃えてください。

Tips ヒアリングメモを読むときは、メモ帳の **表示** で **右端で折り返す** を外すと、下の帯に出る行番号がメモの行と1対1で合います。Ctrl+G で行番号を指定して飛べます。.md をダブルクリックして別のアプリが開くときは、右クリック → **プログラムから開く** → **メモ帳** を選びます。

架空データの扱い

題材の会社、取引先、人物、金額、勤怠と工数のデータは、すべて研修のために作った架空のものです。研修中に、自分の案件の資料や実在のお客様の情報を Claude に渡さないでください。自分の案件で試すときの手順は、Day3 のあとに配る持ち帰り演習集にあります。

Code タブの使い方

研修で使う AI は、Claude デスクトップアプリの Code タブだけです。ターミナル、VS Code、WSL は使いません。Windows の画面写真は載せていないので、ボタンの名前を手がかりに進めてください。講師が同じ操作を画面共有で見せます。

フォルダの開き方

- 1

Claude デスクトップアプリを開き、上の **Code** タブを選びます。
- 2

環境が **ローカル** になっていることを確かめます。
- 3

フォルダを選択 を押し、デスクトップの **idup-handson** を選びます。
- 4

画面の上部に **idup-handson** と出れば開けています。事前準備の Step 3 と同じ操作です。

@ によるファイル指定

入力欄で **@** を打つと、フォルダの中のファイルを指定できます。**@** に続けてファイル名の一部（たとえば「改修」）を入れると候補が出るので、選びます。

このガイドの送る文面には、先頭に **@資料/day1/改修依頼書_KINTAI-2026-03.md** のような行があります。コピーして貼ったままでも Claude はそのファイルを探して読みますが、**@** を打ち直して候補から選ぶと、ファイル名の打ち間違いが起きません。

モードの切り替え

入力欄の下にモード名が出ています。押すと切り替えられます。

モード	Claude のふるまい	研修での使い方
手動	ファイルを書き換える前と、コマンドを実行する前に、そのつど確かめを求めます	Day1 の D1-1 と D1-2 はこれで進めます。Claude が何をしようとしているかを1回ずつ見るためです
編集を受け入れる	ファイルの書き換えは確かめなしで進みます。コマンドの実行は確かめを求めます	D1-3 から切り替えを勧めます。差分の表示で中身を確かめられるので、確かめの回数を減らします
プラン	ファイルを書き換えず、進め方の案だけを返します	作らせる前に手順を見たいときに使います
自動	確かめの多くを省いて進みます	研修では使いません

差分の見方

Claude がファイルを作ったり書き換えたりすると、応答の中に `+12 -1` のような表示が出ます。+ は足した行、- は消した行の数です。押すと、変えた中身を確認られます。

見るのは次の3点です。書き換えたのが頼んだファイルか。`資料/`の中を書き換えていないか。頼んでいない章や行まで消していないか。

コマンド実行の許可

変換ツールや環境チェックのように、Claude がコマンドを実行するときは、実行してよいかを聞かれることがあります。表示されたコマンドを読んでから `許可` を押します。

コマンドの形	扱い
<code>python tools/md_to_docx.py 成果物/day1/… 成果物/day1/…</code> のように、 <code>tools/</code> の変換ツールで <code>成果物/</code> の中に書き出すもの	許可します。 <code>python</code> の代わりに <code>py</code> で始まることもあります
<code>python tools/check_env.py</code>	許可します
ファイルの削除、 <code>資料/</code> への書き込み、ソフトの追加 (<code>pip install</code> など)	許可しません。入力欄で「～はしないでください」と伝え直します

D1-1 文脈の有無による出力の違い

[25min]

個人演習

ねらい

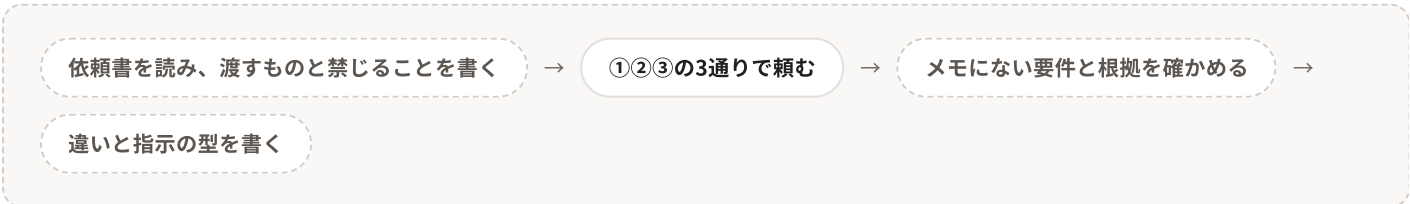
同じ依頼でも、何を渡し、何を禁じるかで、出力がどれだけ変わるかを自分の目で見ます。ここで決める指示の型を、このあとの全演習で使います。

できあがりの目安

- ☐ 成果物/day1/D1-1_比較.md に、①②③の3回分の出力が見出しを分けて並んでいる
- ☐ ①の出力のうち、改修依頼書にもメモにも書いていない要件の数が書いてある
- ☐ ③の根拠のうち2つを、メモ帳で開いて確かめた結果が書いてある
- ☐ 違いが自分の言葉で3点、指示の型が1行で書いてある

使うファイルと作るファイル

区分	ファイル	使い方
使う	資料/day1/改修依頼書_KINTAI-2026-03.md	STEP 1 で読みます。②③で Claude に渡します
使う	資料/day1/ヒアリングメモ_01_第1開発部_田中課長.md 資料/day1/ヒアリングメモ_02_第2開発部_鈴木主任.md 資料/day1/ヒアリングメモ_03_管理部_佐藤さん.md	②③で Claude に渡します
作る	成果物/day1/メモ_D1-1.md	STEP 1 の手書きメモ
作る	成果物/day1/D1-1_比較.md	3回分の出力と、自分で書く違いと指示の型



STEP 1

渡すものと禁じることのメモ

AIを開かない

[5min]

Claude はまだ使いません。改修依頼書をメモ帳で開いて読みます。自分がこの依頼を部下に渡して要件を書かせるなら、何を一緒に渡し、何をするとするかを考えます。

成果物/day1/メモ_D1-1.md に、「渡すもの」を2行、「禁じること」を2行書いて保存します。

手が止まったら開く

「渡すもの」は資料の名前だけでなく、その資料から何を読み取ってほしいかまで書くと、STEP 2 の③で使えます。
「禁じること」は、部下が書いてきた要件で困った経験を思い出すと出てきます。たとえば、打ち合わせで誰も言っていない機能が入っていた、数字が推測で書かれていた、などです。

STEP 2 3通りの依頼

[10min]

モードが **手動** になっていることを確かめてから、①②③の順に同じ会話で送ります。①のあとに②、②のあとに③と、足すものを1つずつ増やします。

① 何も渡さない

Claude に送る文面

ファイルは読まずに、あなたの知識だけで教えてください。
勤怠システムの工数入力に予定工数を足す要件を書いて。
書いた内容は 成果物/day1/D1-1_比較.md を新しく作り、「## ① 何も渡さない」の見出しの下に保存してください。

② 依頼書とメモを渡す

Claude に送る文面

@資料/day1/改修依頼書_KINTAI-2026-03.md
@資料/day1/ヒアリングメモ_01_第1開発部_田中課長.md
@資料/day1/ヒアリングメモ_02_第2開発部_鈴木主任.md
@資料/day1/ヒアリングメモ_03_管理部_佐藤さん.md

この4つを読んで、勤怠システムの工数入力に予定工数を足す要件を書いて。
書いた内容は 成果物/day1/D1-1_比較.md の末尾に「## ② 資料を渡す」の見出しを足して、その下に保存してください。

③ 禁止事項と根拠を足す

Claude に送る文面

@資料/day1/改修依頼書_KINTAI-2026-03.md
@資料/day1/ヒアリングメモ_01_第1開発部_田中課長.md
@資料/day1/ヒアリングメモ_02_第2開発部_鈴木主任.md
@資料/day1/ヒアリングメモ_03_管理部_佐藤さん.md

この4つを読んで、勤怠システムの工数入力に予定工数を足す要件を書いて。
- メモにないことは書かない

- 各要件に、根拠のファイル名と行番号を付ける（例 ヒアリングメモ_01:L23）

- 次の2つも守る

<ここに STEP 1 で書いた「禁じること」2行を貼る>

書いた内容は 成果物/day1/D1-1_比較.md の末尾に「## ③ 禁止と根拠を足す」の見出しを足して、その下に保存してください。

手動モードでは、ファイルを書き換えるたびに確かめを求められます。差分の表示で **D1-1_比較.md** だけが変わっていることを見てから **許可** を押します。

できていれば次へ

☐ **D1-1_比較.md** に ① ② ③ の3つの見出しがあり、それぞれの下に要件が並んでいる

STEP 3 メモにない要件と根拠の確かめ

AIを開かない

[5min]

D1-1_比較.md をメモ帳で開きます。

- ①の要件を上から読み、改修依頼書にも3本のメモにも書いていないものの行末に「【メモにない】」と書き足します。承認の流れ、通知メール、グラフ表示のように、もっともらしいけれど誰も頼んでいないものが混ざっていないかを見ます。
- 【メモにない】の数を数え、①の見出しのすぐ下に「メモにない要件 ○件」と書きます。
- ③から根拠を2つ選びます。その根拠のメモをメモ帳で開き、Ctrl+G で行番号へ飛んで、要件と同じことが書いてあるかを確認めます。
- 結果を末尾に「## 確かめた結果」として書きます。合っていたか、ずれていたか、ずれ方はどうだったか。

メモ帳で書き足したら、保存して閉じてから次へ進みます。開いたまま Claude にこのファイルを書き換えさせると、あとでメモ帳の保存をしたときに Claude の書き換えが消えます。

STEP 4 違いの3点と指示の型

AIを開かない

[5min]

D1-1_比較.md の末尾に「## 違い」を作り、①②③の違いを3点書きます。件数の差より、中身の差を書いてください。どの要件が消え、どの要件が増え、どの書き方が変わったか。

続けて「## 指示の型」を作り、「渡す・禁じる・根拠を求める」を自分の言葉で1行にします。これから先の演習で送る文面は、どれもこの型でできています。

違いの書き方の例を見る

「①は予定工数の承認フローや月次レポートまで書いてきたが、どのメモにも出てこない。③では消えた」

「②は田中課長の言い方と鈴木主任の言い方を混せて1つの要件にしていた。③は根拠が2つ付いたので、2人の言うことが違うのが見えた」

「③の根拠を開いたら、1件は行番号が2行ずれていた。中身は合っていた」

うまくいかないとき

①で Claude がフォルダのファイルを読んで答えた

①の出力に依頼書やメモの言葉が出ていたら、「ファイルは読まずに」が効いていません。①の見出しの下を消してもらい、「ファイルは一切読まずに、一般的な知識だけで」と言い換えて送り直してください。この配布フォルダでは Claude が最初に **CLAUDE.md** と **AGENTS.md** を読むので、何も言わないと資料を探しに行くことがあります。

③でも、メモにない要件が混ざる

「③の要件のうち、根拠の行にその内容が書いていないものを挙げてください」と送ります。挙げたものは、消すか、文末に「(要確認)」を付けさせます。AI はもっともらしい要件ほど、根拠がなくても書きます。消すかどうかを決めるのは自分です。

根拠の行番号が、メモ帳の行番号とずれる

メモ帳の **表示** で **右端で折り返す** を外してから数え直してください。それでもずれるときは、「根拠の行番号を、ファイルを読み直して数え直してください」と送ります。どれだけずれたかは、追加と考察で使います。

ファイルが作られない、別の場所に保存された

差分の表示で、どのファイルが変わったかを見ます。保存先が違ったら「**成果物/day1/D1-1_比較.md**」に保存し直してください」と送ります。手動モードで確かめを求められたのに押し忘れていると、ファイルは作られません。

早く終わった方へ

追加と考察

③の根拠を全部開き、行番号がずれている要件の数を数えます。ずれていたら、なぜずれたのかを考えます。見出しや空行を数えに入れたのか、2行にまたがる発言の1行目と2行目のどちらを指したのか。根拠の付いた出力でも、そのまま信じてよいとは

限りません。

発展課題

③の文面の最後に、次の1行を足して送り直します。

Claude に送る文面

分からないところは、要件を書く前に、私への質問として返してください。

返ってきた質問を読み、ヒアリングで本当に聞くべきものに印を付けます。聞くべき質問が多いほど、メモだけでは要件が決まらないということです。D1-4 の確認事項リストの下書きになります。

D1-2 ヒアリングメモからの要件抽出

[40min]

個人演習

ねらい

3本のメモから、要件を根拠付きで抜き出します。メモ同士で言っていることが食い違う箇所を、要件一覧の中に見える形で残します。

できあがりの目安

- ☐ 成果物/day1/要件一覧.md に要件が25件以上ある
- ☐ 全件に、ファイル名と行番号の根拠がある
- ☐ 「要確認」が3件以上あり、それぞれに誰に何を聞くかが書いてある
- ☐ メモ同士の食い違いが3つ以上、要確認として載っている。食い違いは3本のメモと依頼書の中に5つあります

使うファイルと作るファイル

区分	ファイル	使い方
使う	資料/day1/ヒアリングメモ_01_第1開発部_田中課長.md 資料/day1/ヒアリングメモ_02_第2開発部_鈴木主任.md 資料/day1/ヒアリングメモ_03_管理部_佐藤さん.md	STEP 1 で自分で読み、STEP 2 で Claude に渡します
使う	資料/day1/改修依頼書_KINTAI-2026-03.md	STEP 2 で渡します
作る	成果物/day1/自分の要件候補.md	STEP 1 で自分で書く要件候補
作る	成果物/day1/要件一覧.md	STEP 2 で Claude に作らせ、STEP 3 と STEP 4 で直して確定します



STEP 1

要件候補の手書き

AIを開かない

[10min]

3本のメモをメモ帳で読み、要件の候補を10件以上、成果物/day1/自分の要件候補.md に書きます。1本に3分ずつ使い、最後の1分で見直します。

1行に1件です。行末に、どのメモの何行目かを書きます。2人の言うことが合わないと思った箇所には、行頭に「？」を付けます。

書き方の形（図書館の貸出システムの例。中身は自分で書く）

- 貸出の延長を画面から申し込める（01:L行番号）
- ？ 延長を申し込める期限（01:L行番号 と 03:L行番号 で言い方が違う）
- 返却が遅れた人にメールで知らせる（02:L行番号）

ヒント 食い違いを探す場所

同じ話題が2本以上のメモに出てくる箇所を探し、言っていることが同じかを見比べてください。数字、期限、誰がやるかは、話す人によって言い方がずれやすいところです。

Tips 要件は「～したい」のままにせず、「～できる」「～を表示する」のように、できあがった状態で書くと、あとで Claude の出力と比べやすくなります。

STEP 2 Claude による抽出

[10min]

自分の候補はまだ見せません。先に見せると、Claude がそれに引っぱられて、比べる意味がなくなります。

Claude に送る文面

@資料/day1/ヒアリングメモ_01_第1開発部_田中課長.md
@資料/day1/ヒアリングメモ_02_第2開発部_鈴木主任.md
@資料/day1/ヒアリングメモ_03_管理部_佐藤さん.md
@資料/day1/改修依頼書_KINTAI-2026-03.md

この4つから要件を抜き出して、表にしてください。

列は 要件ID / 要件 / 分類 / 根拠 / 発言者 / 要確認 の6つです。

- 要件IDは REQ-001 から順に振る
- 分類は 機能・非機能・制約・課題 のどれか
- 根拠は「ヒアリングメモ_01:L23」の形で、1件ごとに必ず付ける
- メモ同士で言っていることが食い違う箇所は、要確認の列に「食い違い」と書き、相手のメモの行も根拠に並べる
- 1件に1つのことだけを書く
- メモにないことは書かない

表は 成果物/day1/要件一覧.md に保存してください。

届いた表の件数を見ます。25件に届かないときは、「1件に2つ以上のことが入っている要件を分けてください」と送ります。

STEP 3 自分の候補との突き合わせ

[10min]

ここで初めて、自分の候補を Claude に見せます。Claude には仕分けだけをさせ、どう直すかは自分で決めます。

Claude に送る文面

@成果物/day1/要件一覧.md

@成果物/day1/自分の要件候補.md

自分の要件候補と要件一覧を突き合わせて、次の3つに分けた表を返してください。要件一覧はまだ書き換えないでください。

1. 自分の候補にだけあるもの
 2. 要件一覧にだけあるもの
 3. 同じことを指しているが、中身が食い違うもの
- それぞれ、根拠のメモの行も並べてください。

返ってきた表を見ながら、1件ずつ扱いを決めます。自分の候補にだけあるものは、メモに根拠があれば要件一覧に足します。要件一覧にだけあるものは、根拠の行を開いて確かめます。食い違うものは「要確認」にし、誰に何を聞けば決まるかを書きます。

決めたことを、次の文面の <ここに…> に自分で書いて送ります。

Claude に送る文面

成果物/day1/要件一覧.md を次のとおり直してください。

<ここに、足す要件・消す要件・要確認にする要件と、それぞれ誰に何を聞くかを自分で書いて貼る>

要確認の列は「要確認: 営業部の課長と経理の担当に、請求書の発行日をどちらに合わせるかを確かめる」のように、相手と聞くことを書いてください。

STEP 4 要件一覧の確定

AIを開かない

[10min]

根拠の確かめは自分でやります。Claude に「根拠は正しいか」と聞いても、同じ読み違いをしたまま「正しい」と返ることがあるからです。

- 1 要件一覧から、メモが別々になるように3件を選びます。
- 2 根拠のメモをメモ帳で開き、Ctrl+G で行へ飛びます。要件と同じことが書いてあるか、発言者の列が合っているかを見ます。
- 3 ずれていたら、要件一覧の該当行を自分で直すか、Claude に直させます。
- 4 件数、根拠のない行、要確認の数を数え、できあがりの目安と見比べます。数えるのは Claude に頼んでも構いません。

見直すところ

- ☐ 「～したい」の願望のまま残っている要件がない
- ☐ 1件に2つ以上のことが入っていない
- ☐ 発言者と根拠のメモが合っている
- ☐ 食い違いの要確認に、聞く相手が2人とも書いてある

うまくいかないとき

AI がメモにない要件を足してくる

「根拠の列にある行を読み直し、その行に書いていない要件を挙げてください」と送ります。挙がったものは消すか、要確認にします。「一般的に必要」「通常は」という言い方で入ってくる要件は、たいていメモにありません。

25件に届かない

「1件に2つ以上のことが入っている要件を分けてください」「非機能と制約の要件を、メモからもう一度探してください」と送ります。それでも届かないときは、メモの中の困りごとの発言を拾えているかを見てください。困りごとは、裏返すと要件になります。

表の列がずれる、崩れる

メモ帳で `要件一覧.md` を開き、行の先頭と末尾の `|` の数を見ます。「表の列が6つそろっているかを全行確かめて、ずれた行を直してください」と送ると直ります。要件の中に `|` が入っていると列がずれます。

根拠が「ヒアリングメモ_01」だけで、行番号がない

「根拠に行番号のない要件を挙げ、行番号を付けてください」と送ります。1つの要件が複数の行にまたがるときは「L23-L25」の形で書かせます。

早く終わった方へ

追加と考察

AI が拾わなかった要件は、メモの中でどんな書き方をされていたかを見ます。雑談のような前置き、「ちなみに」で始まる一言、「困ってはいないけど」という否定の形、別の話題の途中に出てくる数字。自分の現場のヒアリングメモで、同じ書き方が出てくる場所を思い浮かべてください。

発展課題

要件に優先度を付けさせ、判断の根拠をメモの発言で示させます。

Claude に送る文面

@成果物/day1/要件一覧.md

各要件に優先度の列を足してください。値は「初期リリース必須」か「次版」のどちらかです。

判断の根拠になったメモの発言を、ファイル名と行番号で示してください。メモに判断の材料がない要件は「要確認」にしてください。

私の考えでは、初期リリースに必ず入れたいのは次の要件です。

<ここに、自分が初期リリース必須だと思う要件IDを3つ書く>

自分の3つと Claude の判断が食い違ったら、どちらの根拠が強いかを比べます。

D1-3 要件定義書のドラフト作成

[35min]

個人演習

ねらい

要件一覧をテンプレートに流し込み、Word の要件定義書にします。分からない箇所は推測で埋めず、「要確認」と確認の相手を書いて残します。

できあがりの目安

- ☐ `成果物/day1/要件定義書_v0.1.docx` が LibreOffice Writer で開ける
- ☐ テンプレートの「書き方」の章を除いて、見出しが全部ある
- ☐ 空欄がない。書けない箇所には「要確認」と、確認する相手が書いてある

使うファイルと作るファイル

区分	ファイル	使い方
使う	<code>テンプレート/要件定義書テンプレート.md</code>	見出しと順番をこのとおりにします
使う	<code>成果物/day1/要件一覧.md</code>	D1-2 で確定したもの
使う	<code>tools/md_to_docx.py</code>	Claude が Word への変換に使います
作る	<code>成果物/day1/メモ_D1-3.md</code>	STEP 1 の手書きメモ
作る	<code>成果物/day1/要件定義書_v0.1.md</code>	Markdown の下書き。これが正本です
作る	<code>成果物/day1/要件定義書_v0.1.docx</code>	変換した Word

Tips この演習から、モードを `編集を受け入れる` に切り替えると進めやすくなります。ファイルの書き換えは確かめなしで進み、変換のコマンドだけ確かめを求められます。中身は差分の表示で見てください。



STEP 1 書きにくい章の見当

AIを開かない

[5min]

テンプレート/要件定義書テンプレート.md をメモ帳で開き、## で始まる見出しを上から見ます。最初の「書き方」の章は、書くときの決まりです。要件一覧の中身で埋まりそうな章と、埋まりそうにない章を分けます。

成果物/day1/メモ_D1-3.md に、一番書きにくい章と、その章を誰に確認するかを2行で書きます。

書き方の例を見る

「非機能要件の性能。『ストレスなく使えること』としか言われていない。今の画面が何秒で開くかを情報システム課に聞く」

STEP 2 Markdown での下書き

[15min]

Claude に送る文面

@テンプレート/要件定義書テンプレート.md

@成果物/day1/要件一覧.md

テンプレートの見出しと順番をそのまま使い、要件一覧の中身を流し込んで、要件定義書を Markdown で書いてください。保存先は成果物/day1/要件定義書_v0.1.md です。

- 「書き方」の章は守る決まりとして読み、書き終えたら消す
- ほかの見出しは消さない。書くことがない章も見出しを残し、「要確認」と書く
- 要件一覧にないことを推測で書いた文には、文末に「(要確認: 確認する相手)」を付ける
- 各要件には、要件一覧の要件IDを残す
- 要件一覧で「食い違い」になっている要件は、どちらかに決めずに両方の言い分を並べる

私が一番書きにくいと思った章と、確認する相手は次のとおりです。この章は特に、推測で埋めずに要確認で残してください。

<ここに STEP 1 のメモ2行を貼る>

できあがったら、差分の表示を開いて章の並びを見ます。テンプレートの見出しが途中で抜けていないか、見出しの名前が勝手に変わっていないかを確認めます。

Tips 要件定義書の正本は Markdown です。Word で直すと Markdown と食い違い、Day2 で Claude に渡すときに古い中身を読ませることになります。直すときは Markdown を直して、変換し直してください。

STEP 3 Word への変換と確認

[10min]

Claude に送る文面

成果物/day1/要件定義書_v0.1.md を tools/md_to_docx.py で Word にして、成果物/day1/要件定義書_v0.1.docx に保存してください。

- 1 コマンドの確かめが出たら、`python tools/md_to_docx.py 成果物/day1/要件定義書_v0.1.md 成果物/day1/要件定義書_v0.1.docx` の形になっているかを見て `許可` を押します。
- 2 エクスプローラーで `成果物` → `day1` を開き、`要件定義書_v0.1.docx` をダブルクリックします。LibreOffice Writer 以外が開くときは、右クリック → `プログラムから開く` → `LibreOffice Writer` を選びます。
- 3 F5 を押すとナビゲーターが開き、見出しの一覧が出ます。テンプレートの見出しと数が合っているかを見ます。

見るところ

- ☐ 見出しが見出しの書式になっていて、ナビゲーターに並ぶ
- ☐ 表に罫線があり、見出しの行に色が付いている
- ☐ `|` や `**` のような記号が、そのまま文字で残っていない
- ☐ 日本語が化けていない

STEP 4 要確認の数え直し

AIを開かない

[5min]

Writer で Ctrl+H を押し、検索に「要確認」と入れて `すべて検索` を押すと、件数が出ます。章ごとの件数を `メモ_D1-3.md` に書きます。

要確認が章の文の半分以上を超えているなら、その章は書き直します。Markdown をメモ帳で開き、要件一覧から自分で書けるところを書き、残りは確認の相手を具体的にします。直したら保存して閉じ、STEP 3 の文面で変換し直します。

要確認が0件の章も疑ってください。メモにない数字や言い切りで埋まっていないかを見ます。

うまくいかないとき

python が見つからない

Claude が「python が見つかりません」と返したら、「py で実行してください」と送ります。`AGENTS.md` にも同じことが書いてあるので、多くは Claude が自分で `py` に切り替えます。どちらも動かないときは、Windows の `設定` → `アプリ` → `アプリの詳細設定` → `アプリ実行エイリアス` で、`python.exe` と `python3.exe` をオフにしてから Claude デスクトップアプリを開き直してください。それでも動かなければ、Python の導入が済んでいないので講師に知らせてください。

変換で文字化けする

Writer で化けているときは、まず元の Markdown をメモ帳で開いて化けていないかを見ます。Markdown が化けていなければ、Writer の書体が Meiryo 以外になっていないかを見ます。Markdown をメモ帳で書き直した場合は、保存するときの **エンコード** が **UTF-8** になっていたかを確認してください。

LibreOffice で開けない

エクスプローラーで docx の大きさが 0 KB なら、変換が失敗しています。Claude の応答にエラーの文が出ていないかを見て、その文を貼って「変換のエラーを直してください」と送ります。ダブルクリックで別のアプリが開くときは、右クリック → **プログラムから開く** で LibreOffice Writer を選びます。

テンプレートの見出しが減った、名前が変わった

「テンプレートの見出しと v0.1 の見出しを並べ、足りないものと名前が違うものを直してください」と送ります。

早く終わった方へ

追加と考察

非機能要件の章に、メモにない数字が入っていないかを見ます。応答3秒、同時接続100人、稼働率99.9%のような数字は、要件定義書らしく見えるので AI が入れがちです。メモにある速さの言い方を、どの数字にして誰に確かめるかを書き出してください。

発展課題

用語集の章を足し、ヒアリングメモで言い方が揺れている語をそろえさせます。

Claude に送る文面

@成果物/day1/要件定義書_v0.1.md
@資料/参考/用語集.md

要件定義書に「用語集」の章を足してください。ヒアリングメモで言い方が揺れている「工数」「作業時間」「稼働」を中心に、同じものを指す言葉を1つにそろえ、本文もそろえた言葉に直してください。
資料/参考/用語集.md と定義が違う語があれば、直さずに要確認にしてください。

D1-4 抜け漏れ探し

[25min]

個人演習

ねらい

AI に頼む前に、自分でデータを開いて要件の抜けを探します。AI の抜け漏れ探しと並べて、どちらが何を拾うかを知ります。

できあがりの目安

- ☐ 自分で見つけた抜けが1件以上、要件定義書 v0.2 と要件一覧に入っている
- ☐ 成果物/day1/確認事項.md に、相手・聞くこと・なぜ の3列で5件以上ある
- ☐ 成果物/day1/要件定義書_v0.2.md と 要件定義書_v0.2.docx がある

使うファイルと作るファイル

区分	ファイル	使い方
使う	資料/データ/勤怠工数CSV_202609.csv	2026年9月の全員の勤怠と工数。LibreOffice Calc で開きます
使う	成果物/day1/要件定義書_v0.1.md	D1-3 で作ったもの
作る	成果物/day1/メモ_D1-4.md	自分で見つけた抜けと、比べた結果
作る	成果物/day1/要件定義書_v0.2.md 成果物/day1/要件定義書_v0.2.docx	抜けを反映したもの。Day2 の入力になります
作る	成果物/day1/確認事項.md	ヒアリングで確かめることの一覧
直す	成果物/day1/要件一覧.md	見つけた抜けを要件として足します。Day3 の対応表に入ります

CSV を開いて抜けを1つ探す

→

Claude に4観点で探させる

→

自分と AI の発見を並べる

→

v0.2 と確認事項を作る

STEP 1 勤怠 CSV からの抜け探し

AIを開かない

[8min]

- 1 エクスプローラーで **資料** → **データ** を開き、**勤怠工数CSV_202609.csv** を右クリック → **プログラムから開く** → **LibreOffice Calc** を選びます。
- 2 **テキストのインポート** の画面が出たら、**文字エンコーディング** を **Unicode (UTF-8)**、区切りのオプションを **コンマ** にして **OK** を押します。
- 3 1行目を選び、**データ** → **オートフィルター** を押すと、列ごとに絞り込めます。
- 4 要件定義書 v0.1 に書いていないことが起きている行を1つ探します。

列	中身
社員ID・表示名	社員ID は S で始まります
日付・勤務区分	1日に複数のプロジェクトを付けた日は、同じ日付の行が複数あります
出勤・退勤・休憩分・実働分	KIN020 で入れた勤怠。実働は分で入っています
PJコード・作業区分・作業分	KIN030 で入れた工数。作業分も分です
備考	入力した人が自由に書いた一言

見つけたら、**成果物/day1/メモ_D1-4.md** に3行で書きます。どの行か（社員ID と日付）、何が起きているか、要件定義書のどの章に書くべきことか。

ヒント

9月の営業日は19日で、9/21・22・23 は休日です。日付で 9/22 を絞り込んでみてください。もう1つは備考の列です。「修正」という言葉が入っている行を探してください。

CSV は保存せずに閉じます。 閉じるときに保存を聞かれたら **保存しない** を選びます。**資料/** の中は書き換ええない約束です。

STEP 2 Claude による抜け漏れ探し

[7min]

自分の発見はまだ見せません。Claude が何を拾うかを、自分の発見と比べるためです。

Claude に送る文面

@成果物/day1/要件定義書_v0.1.md
@資料/データ/勤怠工数CSV_202609.csv

要件定義書の抜け漏れを、例外・運用・権限・データ移行の4つの観点で探してください。

- 観点ごとに表にし、列は 観点 / 抜けていること / 根拠 / 誰に確認するか
- 根拠は、CSV なら社員ID と日付、ヒアリングメモならファイル名と行番号
- 要件定義書はまだ書き換えしないでください

STEP 3 自分の発見との比較

AIを開かない

[5min]

メモ_D1-4.md に、自分の発見と Claude の指摘を並べる表を書きます。

メモに貼る表の形

抜けていること	自分	Claude	要件にするか、運用で済ませるか
---	---	---	---
		○	

自分だけが拾ったもの、Claude だけが拾ったもの、両方が拾ったものを分けたら、どちらが何を拾ったかを1行で書きます。文書を読んで見つかる抜けと、データを開かないと見つからない抜けは、同じではありません。

STEP 4 v0.2 と確認事項の作成

[5min]

反映すると決めた抜けを、自分の言葉で書いてから送ります。Claude の指摘から採るものも、そのまま貼らずに書き直してください。書き直せない指摘は、まだ採る理由が自分の中にない指摘です。

Claude に送る文面

@成果物/day1/要件定義書_v0.1.md
@成果物/day1/要件一覧.md

次の抜けを要件定義書に反映して、成果物/day1/要件定義書_v0.2.md として保存してください。v0.1 は書き換えずに残してください。

<ここに、反映すると決めた抜けを自分の言葉で書いて貼る>

同じ抜けを 成果物/day1/要件一覧.md にも、続きの要件IDで足してください。根拠は、CSV なら社員ID と日付、メモならファイル名と行番号にしてください。

続けて、tools/md_to_docx.py で 成果物/day1/要件定義書_v0.2.docx も作ってください。

最後に、v0.2 の（要確認）を全部拾い、成果物/day1/確認事項.md に表でまとめてください。列は 相手 / 聞くこと / なぜ聞くか の3つです。

できていれば完了

- ☐ 要件定義書_v0.2.docx が Writer で開ける
- ☐ 自分で見つけた抜けが v0.2 に入っている
- ☐ 同じ抜けが 要件一覧.md にも、要件IDと根拠つきで入っている
- ☐ 確認事項.md が5件以上あり、「なぜ聞くか」が空いていない

うまくいかないとき

CSVを開いたら文字化けした

テキストのインポート の画面で 文字エンコーディング が Unicode (UTF-8) になっていなかったはず。いったん閉じて、保存せずに開き直します。ダブルクリックでメモ帳が開いた場合も、右クリック → プログラムから開く → LibreOffice Calc でやり直してください。

LibreOffice で開けない

Calc が起動しないときは、事前準備の Step 5 で tools/sample/確認用.pptx が開けたかを思い出してください。開けていなければ LibreOffice の導入が済んでいません。その間は、CSV をメモ帳で開いて 9/22 の行を Ctrl+F で探しても進められます。

AI がメモにない要件を足してくる

この演習は、メモにない抜けを探すのが目的なので、メモにないこと自体は問題ではありません。見るのは根拠です。CSV の行も、メモの行も示せない指摘は、「どのデータからそう言えますか」と聞き返し、答えられなければ採りません。

v0.1 が書き換えられた

差分の表示で、変わったファイルが 要件定義書_v0.1.md になっていたら、「v0.1 は元に戻し、変更は v0.2 に書いてください」と送ります。

早く終わった方へ

追加と考察

CSV で見つかる汚れを、要件として扱うべきものと、運用で済むものに分けます。休日の出勤は、画面の入力チェックで止めるのか、管理部が月末に確かめるのか。締めあとの修正は、画面で直せるようにするのか、管理部に依頼する運用にするのか。

分けた理由を **メモ_D1-4.md** に書きます。

発展課題

確認事項を、ヒアリングの相手ごとの質問票にさせます。

Claude に送る文面

@成果物/day1/確認事項.md

確認事項を、相手ごとの質問票にしてください。田中課長、鈴木主任、佐藤さん、情報システム課のように相手ごとに見出しを分け、1問ずつ「はい・いいえ」か数字で答えられる聞き方にしてください。

保存先は 成果物/day1/質問票.md です。

「はい・いいえ」で答えられない質問が残ったら、それは会って聞くべき質問です。

よくある質問

演習が時間内に終わりませんでした

途中まででも、ファイルは消さずに残してください。Day2 の最初に、講師が **成果物/合流用/** に要件一覧と要件定義書 v0.2 の合流版を配ります。自分の続きをやりたい方は、研修のあとに同じ手順で進められます。

「許可」を何度も求められて進みません

D1-3 からは、モードを **編集を受け入れる** に切り替えてください。ファイルの書き換えは確かめなしで進みます。変換ツールのようなコマンドの実行は、これまでどおり確かめを求められます。

Claude の返事が途中で止まりました

少し待っても動かないときは、「続けてください」と送ります。ファイルが途中まで書かれていたら、「**成果物/day1/要件一覧.md**」を読み直して、途中から書き足してください」のように、そのファイルの名前を入れて送ってください。

資料フォルダのファイルを書き換えてしまいました

配布 ZIP を別の場所に展開し直し、書き換えたファイルだけを **資料/** に上書きで戻してください。**成果物/** は触らなくて構いません。ZIP は教材サイトの配布物から、いつでもダウンロードできます。

根拠の行番号がメモ帳の行番号と合いません

メモ帳の **表示** で **右端で折り返す** を外してください。それでも合わなければ、Claude が数え違えています。「根拠の行番号を、ファイルを読み直して数え直してください」と送ります。

自分の案件のメモで試してもよいですか

研修中は配布の架空データだけを使ってください。自分の案件で回す手順は、Day3 のあとに配る持ち帰り演習集にあります。社内のルールで AI に渡してよい情報を確かめてから試してください。

Java が分かりませんが、ついていきますか



研修でコードを読む場面はありません。KINTAI が Java で作られているのは題材の背景としてだけで、演習で扱うのはヒアリングメモ、要件定義書、画面設計書、画面のモックです。



上流工程 要件定義・設計 生成AI活用研修 株式会社インフォメーション・ディベロプメント様向け © Givery, Inc.

※題材のサンプル情報サービス株式会社、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。