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

Day1 Claude Codeの基礎とバイブコーディング

Claude Codeを起動して、対話型AIとAIエージェントの違いを手で確かめます。後半は、言葉だけで工数集計の画面を1枚作り、数字を自分で検算します。

2026年11月10日（火） Zoom 個人演習 Claude Code（WSLのUbuntu）

目次

本日のゴール
3日間の全体像
当日の流れ
Zoomでの進め方
始める前の準備
VS Codeで配布フォルダを開く
Claude Codeを最新版にする
Claude Codeに文面を送るとき
演習のメモの置き場所
Claude Codeの基礎
モード
承認の画面
よく使う操作
D1-1 Claude Codeの起動と最初の依頼
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 起動とモードの確認
STEP 2 フォルダの中身の説明
STEP 3 Planモードでの計画
STEP 4 使った量の確認
うまくいかないとき
早く終わった方へ
対話型AIとAIエージェントの違い
AIエージェントとは
道具と権限
コンテキストの窓
AIエージェントでできること
注意点
D1-2 バイブコーディング演習
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 画面に出したいものの3行
STEP 2 Planモードでの計画
STEP 3 ページの作成と表示
STEP 4 言葉だけの修正
STEP 5 検算と承認の振り返り
STEP 6 プロジェクトでの絞り込み
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

今日は、Claude Codeをファイルを読んで書くAIエージェントとして動かし、言葉だけで画面を1枚作ります。作った画面の数字は、AIではなく自分で検算します。

本日の学び

- 対話型AIとAIエージェントの違いを、自分の画面で確かめる
- 承認の画面で、Claudeが何をしようとしているかを読んでから許可する
- AIが作ったものを、元のデータに戻って自分で確かめる

学びを確かめる成果物

- [ ] D1-1 で、配布フォルダの説明と、Planモードの計画が返っている
- [ ] `app/index.html` をブラウザで開くと、社員ごとの今月の工数が表とグラフで出る
- [ ] `メモ_D1-2.md` に、自分で決めた3行と、言葉で直した指示が3つ以上残っている
- [ ] 1人分の合計をCSVで検算し、画面の数字と一致している

3日間の全体像

3日間の演習は、下の3つの力のどれかを練習します。

AIエージェントに任せて確かめる力

Claude Codeは、ファイルを読み、書き、コマンドを動かして作業を進めます。何を任せるか、承認の画面で何を読むか、結果をどう確かめるかは人が決めます。

ハーネスで振る舞いをそろえる力

CLAUDE.md、Subagents、Skills、Hooks、MCPを置くと、同じ依頼でも結果が変わります。どれをいつ使うかを、自分の画面の差で確かめます。

コンテキストを整える力

AIに渡す資料の種類と置き場を決めておくと、毎回説明しなくても前提がそろいます。

回	身に付くこと	作るもの
Day1 Claude Codeの基礎とバイブコーディング（本日）	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	バイブコーディングで作った工数集計の画面1枚
Day2 CLAUDE.md、ハーネス、コンテキスト	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 理想的なハーネスと、ハーネスの作り方	1つのプロジェクトにそろえる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、別テーマの成果物

Day2 とDay3は、回ごとに新しい配布フォルダから始めます。Day1で作った画面は、Day2では使いません。

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
開講	3日間の全体像、Claude Codeの基礎（画面、モード、承認、差分の見方）	説明	[15min]
D1-1	Claude Codeの起動と最初の依頼	個人演習	[15min]
講義	対話型AIとAIエージェントの違い	講義	[15min]
講義	AIエージェントとは	講義	[15min]
デモ	AIエージェントでできること	デモ	[20min]
休憩			[10min]
講義	注意点	講義	[15min]
D1-2	バイブコーディング演習	個人演習	[55min]
閉会	質疑応答、アンケート、閉会	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

事前セットアップで展開した`idcc-day1`をVS Codeで開き、ターミナルを用意します。Claude Codeの起動は、D1-1のSTEP 1で全員そろって行います。

VS Codeで配布フォルダを開く

1. VS Codeを起動し、左下の`><`のアイコンから`Connect to WSL`を選びます。左下に`WSL: Ubuntu-22.04`と出たら、つながっています。`WSL: Ubuntu`と出ていたら、つながった先は別の環境です。このPCには`Ubuntu`という環境も入っていますが、研修で使うのは`Ubuntu-22.04`だけです。接続を閉じて、`Connect to WSL using Distro...`から`Ubuntu-22.04`を選び直してください。

2. `ファイル` → `フォルダーを開く` で `/home/ユーザー名/idcc-day1` を開きます。左のエクスプローラーに `data` と `index.html` が見えれば合っています。

3. `ターミナル` → `新しいターミナル` で、画面の下にターミナルを開きます。プロンプトの末尾が `~/idcc-day1$` になっていることを確かめます。

4. ターミナルで `echo $WSL_DISTRO_NAME` と打ち、 `Ubuntu-22.04` と返ることも確かめます。

> **Tips** `idcc-day1` が無いときは、事前セットアップのStep 7で展開してください。講師がZoomのチャットで手順を案内します。

Claude Codeを最新版にする

ターミナルに打つコマンド

```
claude update
```

`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。

Claude Codeに文面を送るとき

> **Tips** このガイドの送る文面は、 `コピー` を押して入力欄に貼ります。改行を含む文面も、そのまま貼れます。入力欄で自分で改行するときはCtrl+Jです。Enterだけを押すと送信されます。

> **Tips** VS Codeのターミナルで Claude Codeを動かすと、VS Codeの拡張機能が自動で入り、ファイルの変更の提案がVS Codeの差分のタブで開くことがあります。どちらで見ても中身は同じです。承認はターミナルの側で行います。

演習のメモの置き場所

AIを開かずに書くメモは、VS Codeで `idcc-day1` の中に作ります。左のエクスプローラーで右クリック → `新しいファイル` を選び、 `メモ_D1-2.md` のように演習番号を付けた名前にします。保存はCtrl+Sです。

04

Claude Codeの基礎

開講の [15min]で、画面の読み方と、よく使う操作を見ます。研修中に迷ったら、この節に戻ってください。

モード

Shift+Tabを押すたびに、モードが切り替わります。いまのモードは、入力欄の下に出る表示で分かります。研修で主に使うのは、通常のモードとPlanモードの2つです。書き換えの承認が続いて手が止まるときは、accept editsを使ってかまいません。

モード	入力欄の下の表示	ファイルの書き換え	コマンドの実行
通常	`manual mode on`	1件ずつ承認する	1件ずつ承認する
編集の自動承認	`accept edits on`	承認なしで進む	作業フォルダの中のmkdir、touch、cp、mv、rm、sedは承認なし。ほかのコマンドと`.claude/`の中への書き込みは、1件ずつ承認する
Plan	`plan mode on`	しない。計画だけを出す	読むための操作だけ
自動	`auto mode on`	承認なしで進む	多くを承認なしで進める。研修では使わない

表示はmanual、accept edits、plan、autoの順に変わり、manualに戻ります。autoは、アカウントの設定によっては出ません。

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。

承認の画面

通常のモードでは、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。ファイルの変更なら、ファイル名と、差分の赤と緑の行を読みます。コマンドのときに読むのは、コマンドの全文と、どのフォルダで動くかです。

研修では、内容を読んで問題がなければ`1. Yes`を、違うと思ったら`No`を選びます。Noの番号は画面によって3番か4番に変わるので、番号ではなく文言を読んで選んでください。`Yes, and`から始まる選択肢は、同じ種類の操作を以後は確かめずに通すので、研修では選びません。

`Yes, and switch to accept edits`を選ぶとaccept editsに切り替わり、ファイルの書き換えに加えて、フォルダの中のrmやmvも聞かれずに進みます。

承認の画面	選択肢	研修で選ぶもの
コマンド（4択の画面）	1. Yes／2. Yes, and don’t ask again for …／3. Yes, and switch to auto mode／4. No	1. Yesか4. No
コマンド（3択の画面）	1. Yes／2. Yes, and switch to auto mode／3. No	1. Yesか3. No
ファイルの作成と書き換え	1. Yes／2. Yes, and switch to accept edits …／3. No	1. Yesか3. No

承認の画面	選択肢	研修で選ぶもの
Planモードの計画	1. Yes, and use auto mode／2. Yes, manually approve edits／3. Tell Claude what to change	2番か3番

コマンドの画面の選択肢の数は、コマンドの形によって3つか4つになります。4択の画面の2番の文言は、版や操作によって `Yes, and always allow access to …` などに変わります。どの画面でも、`Yes, and` から始まる選択肢は選びません。

Planモードの計画の画面は、1番のYes, and use auto modeが選ばれた状態で出ます。Enterだけを押すと自動のモードで進むので、研修では2. Yes, manually approve editsを選んでください。

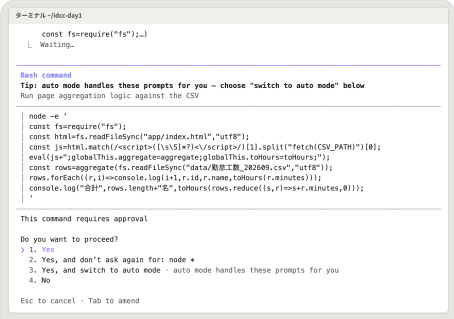

図1：Claudeがコマンドを動かす前に出る画面。動かすコマンドの全文が、点線の枠の中に出る

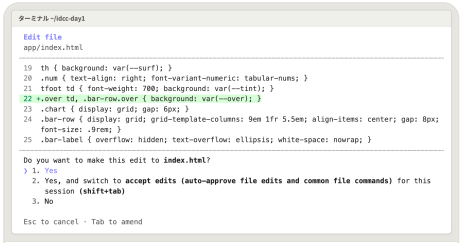

図2：ファイルの書き換えの承認の画面。+の付いた緑の行が足される行

よく使う操作

操作	キーとコマンド	使う場面
モードを切り替える	Shift+Tab	計画だけを出させたいときと、通常に戻すとき
応答を止める	Esc	Claudeが違う方向に進み始めたとき
会話を空にする	`/clear`	演習が変わったとき。前の会話を引きずらず、利用量も抑えられる
コンテキストの使い具合を見る	`/context`	会話が長くなったとき
利用量を見る	`/usage`	研修の始めと、長い作業のあと
終了する	`/exit`	研修の終わり

05

D1-1 Claude Codeの起動と最初の依頼

[15min] 個人演習

この演習の [15min]は、必須のSTEP 1〜4の合計です。

ねらい

いきなり作らせずに、まず中身を説明させ、次に計画だけを出させます。Claudeがファイルを自分で開いて答えるところを見て、対話型AIとの違いを確かめます。

できあがりの目安

- ☐ Claude Codeが起動し、入力欄の下に`manual mode on`と出ている
- ☐ 説明に出たファイルが、VS Codeのエクスプローラーのファイルと一致している
- ☐ Planモードで計画が返り、実行せずに終えている。ファイルは1つも変わっていない
- ☐ `/context`と`/usage`を開いて、使った量を見た



図3：計画が返り、実行するかを聞かれた状態。ここでEscを押して終える

使うファイルと作るファイル

区分	ファイル	使い方
使う	`CLAUDE.md`、`はじめにお読みください.md`、`index.html`、`data/`	Claudeに読ませて、説明させる。書き換えない

起動とモードの確認 → **フォルダの中身の説明** → **Planモードでの計画** → **使った量の確認**

この演習は、1つの会話で最後まで進めます。STEPの文面は前のSTEPの続きとして送るので、途中で`/clear`を打たないでください。演習が変わったら、`/clear`で会話を空にしてから始めます。

STEP 1 起動とモードの確認

[3min]

VS Codeのターミナルで、プロンプトの末尾が `~/idcc-day1$` になっていることを確かめてから、Claude Codeを起動します。

ターミナルに打つコマンド

```
claude
```

1. このフォルダを信頼するかを聞かれたら、↓キーで **Yes, I trust this folder** を選んでEnterを押します。最初はNoの側が選ばれているので、Enterだけを押すと終了します。
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。
3. 起動できたら、Zoomのチャットに「起動」と書いてください。

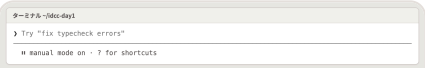

図4：起動した直後の入力欄。下にmanual mode onと出ている

STEP 2 フォルダの中身の説明

[5min]

Claudeに送る文面

```
このフォルダの中身と、それぞれのファイルの役割を説明してください。ファイルは書き換えないでください。
```

Claudeはファイルを開くだけなら聞かずに進め、ファイルを探すコマンドを動かす前には承認の画面を出すことがあります。コマンドを読み、ファイルを書き換えない内容なら1. Yesを選びます。

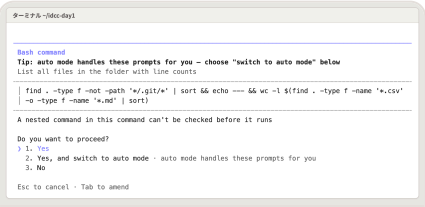

図5：Claudeがファイルの一覧を作る前に出した画面。findとwcをつないだコマンドで、ファイルを数えるだけなので1. Yesを選ぶ

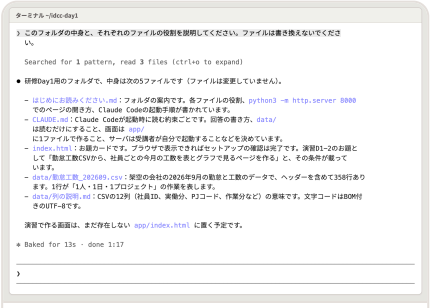

図6：返ってきた説明。フォルダのファイルと役割が並ぶ

返ってきた説明を、VS Codeのエクスプローラーと1つずつ突き合わせます。説明に出てこないファイルや、無いファイルの説明がないかを見てください。応答が短いのは、`CLAUDE.md` に「応答は要点だけ、10行以内」と書いてある

ためです。`CLAUDE.md`の働きはDay2で扱います。

STEP 3 Planモードでの計画 [5min]

Shift+Tabを押して、入力欄の下に`plan mode on`と出る状態にしてから送ります。

Claudeに送る文面

お題カードの index.html の見出しの色を変えるとしたら、どう進めるかの計画を立ててください。

計画の前に、Claudeが「どの見出しか」「何色か」を選択肢で聞いてくることがあります。↑↓キーで選んでEnterを押し、最後に**Submit answers**を選びます。



図7：計画の前にClaudeが聞いてきた画面。選択肢を選んでEnterを押す

1. 計画が返ったら、変えるファイルと、進める順番が書いてあるかを読みます。
2. 実行するかを聞かれたら、Escを押して実行せずに終えます。1. Yes, and use auto modeが選ばれた状態で出るので、Enterは押さないでください。
3. Shift+Tabを何回か押して、入力欄の下を`manual mode on`に戻します。
4. VS Codeのエクスプローラーで、`index.html`が変わっていないことを確かめます。

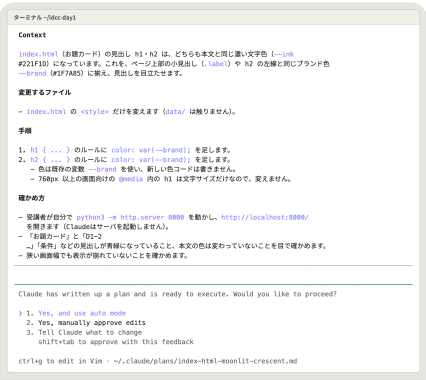

図8：見出しの色を変える計画と、実行するかを聞く画面。ここでEscを押す

STEP 4 使った量の確認 [2min]

入力欄に打つコマンド

/context

コンテキストの窓の、どれだけを使ったかが出ます。右のMessagesが、この会話で使った分です。数字は、使うモデルと会話の長さで変わります。Escで閉じます。

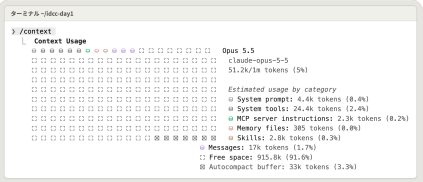

図9：`/context`の表示。左の升目と右の内訳で、使った量が分かる

入力欄に打つコマンド

/usage

研修用の組織の利用量が出ます。表示はアカウントの契約で変わります。見たらEscで閉じ、最後に`/clear`で会話を空にしてください。D1-2は、この空の会話から始めます。

うまくいかないとき

claudeと打っても起動しない

`command not found`と出るときは、VS Codeの左下が`WSL: Ubuntu-22.04`になっているかを見てください。`WSL: Ubuntu`と出ていたら、Claude Codeが入っていない別の環境を開いています。接続を閉じて`Ubuntu-22.04`を選び直してください。Windows側のターミナルで打っている可能性もあります。それでも起動しないときは、Zoomのチャットで知らせてください。

ログインを求められる

事前セットアップのStep 5と同じ手順で、研修用の組織に招待されたアカウントでログインしてください。

Planモードにならない

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`の順に変わります。`plan mode on`が出るまで押してください。

計画のあとにEnterを押して、書き換えが始まった

Escを押して止めます。変わったファイルは、入力欄に`/rewind`と打ち、戻したい時点を選ぶと元に戻せます。どこまで戻っ

たかは、VS Codeで`index.html`を開いて確かめてください。

早く終わった方へ

追加と考察

D1-1 のSTEP 2の文面を、ブラウザのチャットのAIに送ったら、どう返ってくるかを考えてみてください。フォルダの中身を貼らないと、答えようがありません。

発展課題

D1-1+ CSVの列の意味を説明させ、自分でCSVを開いて合っているかを確かめます。

Claudeに送る文面

data/勤怠工数_202609.csv の各列の意味を、1列1行で説明してください。値の例も1つずつ付けてください。

返ってきた説明を、VS CodeでCSVを開いて見比べます。特に`実働分`と`作業分`の違いが、`data/列の説明.md`と合っているかを確かめてください。

06

対話型AIとAIエージェントの違い

講義では、同じ依頼をブラウザのチャットとClaude Codeに出して、返ってくるものの違いを見せます。

観点	対話型AI	AIエージェント
例	ブラウザで使うチャットのAI	Claude Code
返すもの	答えの文章とコード	ファイルの変更と、コマンドの実行結果
手を動かす人	人がコードをコピーして、ファイルに貼る	AIがファイルを読み、書き、コマンドを動かす
見ている範囲	会話に貼ったものだけ	開いたフォルダの中のファイル
人がすること	答えを読んで、使うかを決める	承認の画面を読み、結果を確かめる

D1-1 で、Claudeはフォルダの中のファイルを自分で開いてから答えました。対話型AIなら、人がファイルの中身を貼るところです。

07

AIエージェントとは

AIエージェントは、考える、道具を使う、結果を見る、を目的が済むまで繰り返します。1回の依頼の中で、ファイルを開き、足りなければ別のファイルを探し、書いたあとに動かして確かめます。

道具と権限

道具	Claude Codeの中の名前	通常のモードでの承認
ファイルを読む、探す	Read、Glob、Grep	承認なしで動く
ファイルを書く、直す	Write、Edit	1件ずつ承認する
コマンドを動かす	Bash	1件ずつ承認する。全文を読んでから許可する
Webを調べる	WebSearch、WebFetch	承認を求められたら、調べる先を見てから許可する

accept editsとPlanのモードで承認がどう変わるかは、「Claude Codeの基礎」のモードの表にまとめてあります。

コンテキストの窓

Claudeが一度に覚えておける量には上限があり、これをコンテキストの窓と呼びます。読んだファイルも会話も、すべて窓の中身です。窓が埋まると古い内容が要約され、前に伝えた細かい指示が抜けやすくなります。

`/context` で使い具合を見て、演習が変わったら `/clear` で空にしてください。Day2では、毎回の説明を減らすために、資料をフォルダに置いて読ませる方法を扱います。

08

AIエージェントでできること

講師がClaude Codeで、次の5つを順に見せます。受講者は見るだけです。

デモ	見るところ
既存コードの読解	Claudeがどのファイルを、どの順に開いたか
複数ファイルの改修	1つの依頼で、差分が複数のファイルに出ること
テストを流して直すまでの自走	テストが失敗したあと、人が何も打たずに直して流し直すところ
ブラウザの操作	画面を開いてボタンを押し、結果を読んで次の手を決めるところ
Excelの表の作成	表のファイルを作り、開いて中身を確かめるところ

どのデモも、道具を使って結果を見るループの繰り返しです。デモの途中で、承認の画面に何が出ているかにも目を向けてください。

09

注意点

後半の演習に入る前に、次の7つを押さえます。D1-2のSTEP 5で、このうち2つを自分の画面で確かめます。

注意点	研修で守ること
もっともらしい誤り	画面の数字や説明は、元のデータに戻って確かめる
承認を読まずに押す	コマンドの全文と、変更するファイルの名前を読んでから許可する
機密情報と個人情報	自分の案件の資料、顧客のデータ、パスワードやキーを入れない。研修では架空のデータだけを使う
外部の文書やWebに書かれた指示	読ませた文書の中に指示が書かれていると、AIがそれに従うことがある。外から持ち込んだ文書を読ませたあとの承認は、特に丁寧に読む
危険なコマンド	`rm -rf`、`sudo`、外へデータを送るコマンドは許可しない。迷ったら止めて、講師に聞く
使用量の上限	研修用の組織の利用枠を全員で使う。同じ依頼を何度も送り直さない
生成物の責任	AIが作ったものを使うかは人が決め、その責任も人が持つ

外部の文書に書かれた指示にAIが従ってしまう攻撃を、プロンプトインジェクションと呼びます。

10

D1-2 バイブコーディング演習

[55min] 個人演習

この演習の [55min]は、必須のSTEP 1〜5の合計です。STEP 6は発展で、合計 [10min]です。この演習の枠には入っていません。

ねらい

コードを自分で書かずに、言葉だけで動く画面を作ります。最後に画面の数字を元のデータで検算し、AIの出力を人が確かめる手順を自分の手で通します。

押さえること

完成度は問いません。表が出れば達成で、そこから言葉で直す往復を3回は通してください。1回の指示は短くします。

できあがりの目安

- ☐ ブラウザで `http://localhost:8000/app/` を開くと、12名分の今月の工数が表とグラフで出る
- ☐ `メモ_D1-2.md` に、自分で決めた3行と、言葉で直した指示が3つ以上残っている
- ☐ 1人分の合計をCSVで検算し、画面の数字と一致している。一致しないときは、どこが違ったかがメモに書いてある



図10：できあがりの例。12名の作業時間が表と横棒グラフで出て、合計の行がある。見た目は人によって違います

使うファイルと作るファイル

区分	ファイル	使い方
使う	`data/勤怠工数_202609.csv`	画面で集計するデータ。書き換えない
使う	`data/列の説明.md`	列の意味
作る	`メモ_D1-2.md`	自分で決めた3行と、直した指示を残す

区分	ファイル	使い方
作る	`app/index.html`	Claudeに作らせる画面

画面に出したいものの3行 → Planモードでの計画 → ページの作成と表示 → 言葉だけの修正 → 検算と承認の振り返り

この演習は、1つの会話で最後まで進めます。STEPの文面は前のSTEPの続きとして送るので、途中で`/clear`を打たないでください。演習が変わったら、`/clear`で会話を空にしてから始めます。

STEP 1 画面に出したいものの3行

AIを開かない [5min]

ブラウザでお題カードの`http://localhost:8000/`を開き、条件を読みます。サーバが止まっているときは、事前セットアップのStep 7のとおりに動かしてください。

VS Codeで`メモ_D1-2.md`を作り、画面に出したいものを3行で書きます。1行目は誰が何のために見るか、2行目は表の列、3行目はグラフの形です。



図11：お題カード。D1-2の条件と、データの列の説明が書いてある

別の題材での3行の例を見る

総務の担当者が、会議室の使われ方の偏りを見るために使う

表の列は、会議室名、今月の利用回数、利用時間の合計

利用時間の合計を、会議室ごとの横向きの棒グラフにする

会議室の予約という別の題材での例です。工数の画面の3行は、自分で考えて書いてください。

STEP 2 Planモードでの計画

[10min]

Shift+Tabで入力欄の下に`plan mode on`を出してから、文面を送ります。`<ここに…>`は、STEP 1の3行に置き換えます。

Claudeに送る文面

メモ帳で<ここに…>を埋めてから貼る

お題カードの index.html と data/列の説明.md を読んで、次の3行を満たす画面の計画を立ててください。作るのは app/index.html の1ファイルで、HTML と JavaScript だけで書き、外部のライブラリは使いません。

<ここに STEP 1 で書いた3行>

返ってきた計画で、次の4つを読みます。

- ☐ 作るファイルの名前と場所が`app/index.html`か
- ☐ CSVのどの列を、どう足して工数にするか。分を時間に直す計算も入っているか
- ☐ 外部のライブラリを使っていないか
- ☐ 表とグラフの両方があるか

1. 直したい点を1つ選びます。実行するかを聞かれた画面で、↓キーで3. Tell Claude what to changeを選んでEnterを押します。入力欄に戻ったら、直す点を1文で送ってください。
2. 直った計画が返ったら、↓キーで2. Yes, manually approve editsを選んでEnterを押します。モードは`manual mode on`に変わり、ファイルを作る前に承認の画面が出ます。



図12：直す点を伝えたあとの計画。確認方法に、検算の手順まで書いてある

STEP 3 ページの作成と表示

[15min]

ファイルを作る前に、承認の画面が出ます。ファイル名が`app/index.html`になっているかを見てから1. Yesを選んでください。



図13：ファイルを作る前の承認の画面。上に、作るファイルの中身が出る

ファイルを作ったあと、Claudeが自分で確かめようとして、nodeや別のコマンドの承認を求めることがあります。研修のPCにはNode.jsが入っていません。nodeのコマンドはNoを選び、「Node.jsは入っていないので、確かめは自分でブラウザとCSVで行います」と送ってください。

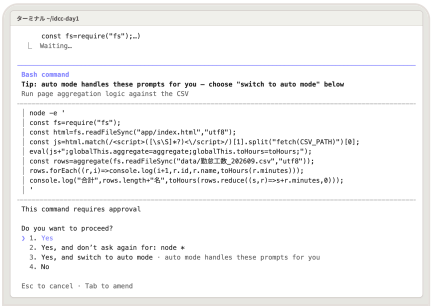

図14：Claudeがnodeで集計を確かめようとしたときに出た画面。研修のPCではNoを選ぶ

> **サーバは自分で動かします。**Claudeが `python3 -m http.server` などのサーバを起動するコマンドの承認を求めてきたら、止める側の選択肢を選び、「サーバは自分で動かします」と送ってください。サーバのコマンドは終わらないので、Claudeの会話が止まります。

ファイルができたら、VS Codeのターミナルの右上の `+` で2つ目のターミナルを開き、サーバを動かします。お題カードを見たときのサーバが動いていれば、そのままで構いません。

2つ目のターミナルに打つコマンド

```
cd ~/idcc-day1
python3 -m http.server 8000
```

Windowsのブラウザで次のURLを開きます。表とグラフが出れば、このSTEPは済みです。

ブラウザのアドレス欄

```
http://localhost:8000/app/
```



図15：ブラウザで開いた画面の例。12名の作業時間が表とグラフで出ている。見た目は人によって違います

STEP 4 言葉だけの修正 [15min]

ブラウザの画面を見て、直したいところをClaudeに言葉で頼みます。1回の指示は1文にして、3回直してください。ファイルを書き換える承認の画面では、差分の赤と緑の行を見てから許可します。

1回の指示で、書き換えの承認が何回か続くことがあります。撮影したときは、1回の指示で7回続きました。毎回、差分を見てから1. Yesを選んでください。

直すたびにブラウザでCtrl+Shift+Rを押し、画面を読み込み直します。送った指示は、`メモ_D1-2.md` に1行ずつ書き足してください。

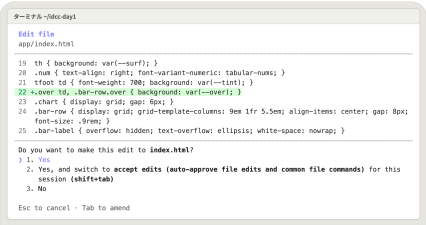

図16：「160時間を超える人の行と棒に、薄い色を付けてください」と送ったときの書き換えの承認の画面



図17：直したあとのブラウザの画面。160時間を超える松本の行に色が付いた

指示の例を見る −

- グラフの棒の右に、時間の数字を出してください。
- 160時間を超える人の行に、薄い色を付けてください。
- 表の上に、12名の合計時間と平均時間を出してください。

> **Tips** 思ったとおりに直らないときは、指示を長くせずに、いまの画面のどこが、どう違うかを1文で伝え直してください。

> **Tips** 書き換えの承認が続いて手が止まるときは、Shift+Tabで入力欄の下を`accept edits on`にして構いません。このモードでは、ファイルの書き換えと、フォルダの中のmkdir、cp、mv、rmなどのコマンドが聞かれずに進みます。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。STEP 5の前に`manual mode on`に戻してください。

STEP 5 検算と承認の振り返り

AIを開かない [10min]

画面の数字が正しいかを、AIに聞かずに自分で確かめます。1人を選び、その人の`作業分`をCSVで足して60で割ります。2つ目のターミナルでCtrl+Cを押してサーバを止め、次を打ってください。`田中`のところは、選んだ人の表示名に変えてください。

2つ目のターミナルに打つコマンド

```
cd ~/idcc-day1
awk -F, '$2=="田中" {s+=$11} END {print s/60}' data/勤怠工数_202609.csv
```

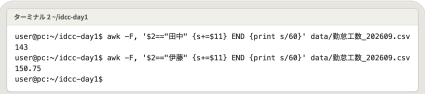

図18：2つ目のターミナルで、田中と伊藤の作業時間を検算した画面。画面の143.00と150.75に一致する

出た数字と、画面の同じ人の数字を比べます。画面が小数第1位までのときは、四捨五入して比べてください。合っていれば、`メモ_D1-2.md`に「検算 一致」と、選んだ人と数字を書きます。合わなければ、どちらがいくつだったかを書いてください。

最後に、Claudeの会話を上へさかのぼり、この演習で許可した操作を数えます。ファイルの作成と書き換え、コマンドの実行のそれぞれが何回あったかを、メモに1行で書いてください。

> **Tips** `awk`は、表の形のテキストを列ごとに計算するUbuntuのコマンドです。`-F,`で区切りの文字をカンマにし、2列目の表示名が一致する行の11列目の作業分を足しています。サーバを止めずに試すときは、もう1つターミナルを開いてください。

ここまでが必須です

STEP 1〜5が終わると、この演習の必須はすべて済みます。ここから先は発展で、合計 [10min]です。時間が残った方が進めてください。

講師は発展を待たずに、次の枠へ進みます。

STEP 6 プロジェクトでの絞り込み

発展 [10min]

D1-2+画面に、プロジェクトで絞り込むボタンを足します。

Claudeに送る文面

表とグラフの上に、PJコードごとのボタンを並べてください。ボタンを押すと、そのプロジェクトの作業分だけで表とグラフを作り直します。「すべて」のボタンで元に戻します。

絞り込んだあとの数字も、STEP 5のコマンドに `&& $9=="PJ-2026-011"` のように条件を足せば検算できます。

うまくいかないとき

ブラウザに何も出ない、または表が空のまま

アドレス欄が `http://localhost:8000/app/` になっているかを見てください。ファイルをダブルクリックして `file://` で開くと、CSVを読み込めません。サーバは `app` の中ではなく、`~/idcc-day1` で動かします。

それでも出ないときは、ブラウザでF12を押し、`Console` の赤い行をコピーして、次の文面に貼って送ってください。

Claudeに送る文面 メモ帳で<ここに…>を埋めてから貼る

ブラウザで app/index.html を開くと表が空です。コンソールに次のエラーが出ています。原因を調べて直してください。

<ここにコンソールの赤い行>

Claudeがnodeのコマンドの承認を求めてきた

研修のPCにはNode.jsが入っていないので、Noを選びます。続けて「Node.jsは入っていないので、確かめは自分でブラウザとCSVで行います」と送ってください。Yesを選んだ場合も、コマンドが見つからないと返るだけで、ファイルは壊れません。

Claudeがサーバを起動して、会話が止まった

Escを押して止めます。動いているサーバは、そのターミナルでCtrl+Cを押すと止まります。サーバは自分で2つ目のターミナルで動かすとClaudeに伝えてから続けてください。

直したのに、ブラウザの画面が変わらない

ブラウザが古い画面を覚えています。Ctrl+Shift+Rで読み込み直してください。

画面の数字が検算より大きい

工数の元にする列が違っている可能性があります。`data/列の説明.md` で、`実働分` と `作業分` の違いを読んでください。どの列を足しているかをClaudeに聞き、計画のときに読んだ列と比べます。

日本語が`?`や記号に化ける

ターミナルで`locale charmap`を打ち、`UTF-8`が返るかを見てください。返らないときは、事前セットアップのStep 4を行ってから、Claude Codeを起動し直します。

ポート8000が使用中と出る

別のターミナルで、前に動かしたサーバが残っています。そのターミナルでCtrl+Cを押して止めてください。

早く終わった方へ

追加と考察

STEP 5 の検算を、もし行わなかったらどうなっていたかを考えてください。画面がきれいでも、足している列が違えば数字は誤ります。AIは、もっともらしい画面を作るのが得意です。

発展課題

D1-2+は、上のSTEP 6です。早く終わった方は、STEP 6のあとに、グラフの色やレイアウトを自由に直してみてください。

11

よくある質問

Claudeが英語で答えます

「日本語で答えてください」と送ってください。毎回の言い直しを減らす方法は、Day2のCLAUDE.mdで扱います。

`/usage`で、上限に近いという表示が出ました

Zoomのチャットで講師に知らせてください。研修用の組織の利用枠を全員で使っています。

Claudeの応答が長くて、読み切れません

「要点だけ5行で」と送ると短くなります。Day2の配布フォルダには、応答を短くさせる指示があらかじめ書いてあります。

作った画面を手元に残せますか

`~/idcc-day1/app/index.html`は普通のファイルです。Windowsのエクスプローラーのアドレス欄に`\\wsl$\Ubuntu-22.04\home\ユーザー名\idcc-day1`と打つと開けるので、コピーして残せます。

研修のあともClaude Codeを使えますか

研修で使うアカウントは、研修のために用意した組織のものです。研修のあとの使い方は、貴社のご担当者の案内に従ってください。

© 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。