

事前セットアップガイド

AIカスタム研修 / 2026年9月29日（火）・10月6日（火）

Day1 の前日までに、Step 0 から Step 4 を終えてください。所要はおよそ30分です。当日は準備の時間を取らずに演習から始めます。

当日使う2つのツールと役割分担

Claude Desktop

この中の Claude Code で演習フォルダを開いて動かします。設定ファイルとスクリプトを書き、既存コードを調べる役です。

GitLab

演習用にこちらでご用意済みです。変更を出し、AIのレビューを受け取る場所です。演習コードの実行とテストもこの上で行うため、お手元に PHP や Docker を入れる必要はありません。

Tips : 当日の手順は [ハンズオンガイド](#) に Day1 と Day2 に分けて載せています。配布物一式と座学スライドは [教材ページの配布物](#) からダウンロードできます。

Step 0 準備物の確認

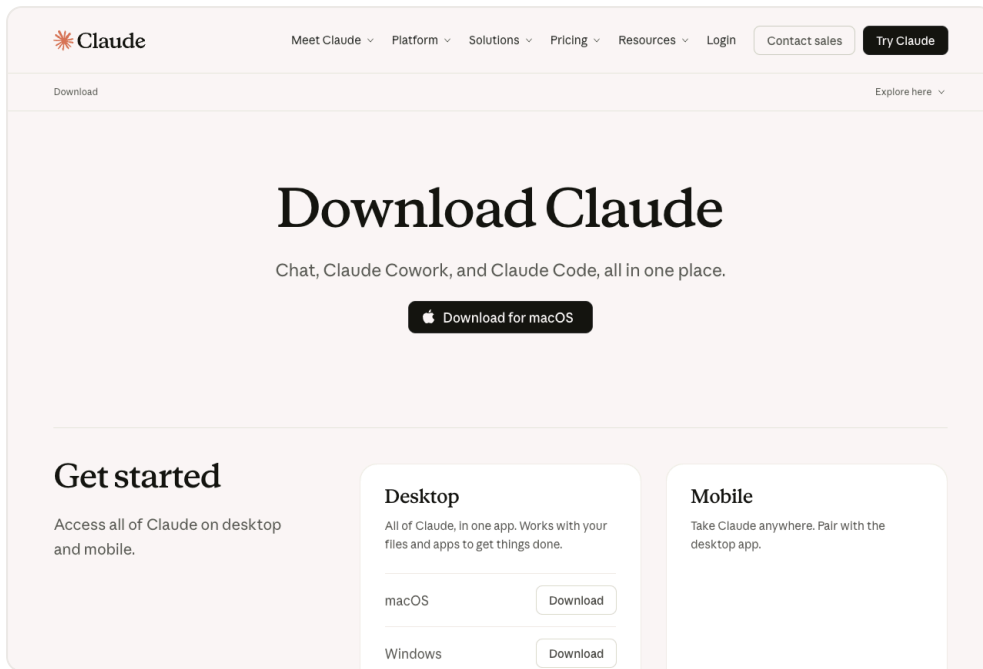
STEP 0 PCに入っているものを表で確認する

手元の PC に次の4つがそろっていれば、Step 1 へ進めます。足りないものがあれば、右の列の手順で導入してください。導入に社内の申請が要る場合もありますので、この表だけは早めに確認してください。

準備物	確認のしかた	入っていないとき
Windows の PC	当日と同じ端末を使います	Mac の方も手順は同じです。画面の見た目だけが少し変わります。スタートメニューはアプリケーションフォルダと読み替えてください
VSCode	スタートメニューで「Visual Studio Code」が見つかる	code.visualstudio.com から Windows 版を導入します。日本語表示するには、左の拡張機能から Japanese Language Pack を入れます。普段お使いのエディタと Git クライアントで代用しても構いません。演習で VSCode を使うのはファイルを読む画面と Git を操作する画面としてです
Git	Step 2 で演習リポジトリを取り込めれば入っています	git-scm.com の Git for Windows を導入します。導入画面の選択肢は全部そのまま進めて構いません。Mac の方は、ターミナルで <code>git -version</code> を実行してバージョンが表示されれば入っています
Claude Desktop	アプリを起動すると、ログイン済みの画面が開く	下の「Claude Desktop の導入」のとおりに入れます

Claude Desktop の導入

研修では Claude Desktop の中の Claude Code を使います。ターミナルの操作は要りません。claude.ai/download を開き、Windows 版のインストーラをダウンロードして実行してください。Mac の方は同じページの macOS 版を選びます。



ダウンロードページです。画面は Mac で撮ったもので、中央の黒い **Download for macOS** は Mac 用です。Windows の方は、下の **Desktop** の欄の2行目、**Windows** の行の右にある **Download** を押してください。Mac の方は1行目の **macOS** の行です

- 1 ダウンロードしたインストーラを実行します。画面の指示とおりに進めるだけで終わります。
- 2 インストールが終わったらアプリを起動し、会社で契約しているアカウントでサインインしてください。
- 3 サインインできたら、画面上部の **Code** タブを開きます。ここが当日使う画面です。

このワークスペースを信頼しますか？

Claude Codeはこのフォルダ内のファイルを読み取り、書き込み、または実行する場合があります。このワークスペースを信頼する場合のみ、続行してください。

/Users/coelaqanth_006/Desktop/dl-training-app

macOSがこのフォルダーへのアクセス許可を求める場合があります。Claude Codeをここで起動できるよう、プロンプトで「許可」をクリックしてください。

詳細については、[セキュリティガイド](#)をご覧ください。

実行を許可:

.claude/settings.json

キャンセル

ワークスペースを信頼

演習フォルダを初めて開くと出る確認です。 `.claude/settings.json` を読み込んでよいかを聞いているので、**ワークスペースを信頼** を押してください。これを押さないと、フックと拒否の規則が読まれません。画面は Mac で撮ったもので、Windows ではパスが `C:\Users\...` の形になり、macOS のアクセス許可についての段落は出ません

Tips：アカウントは会社で契約しているものをお使いください。個人のアカウントで研修を受けると、演習リポジトリの内容が個人の履歴に残ります。

すでにターミナルで Claude Code をお使いの方へ

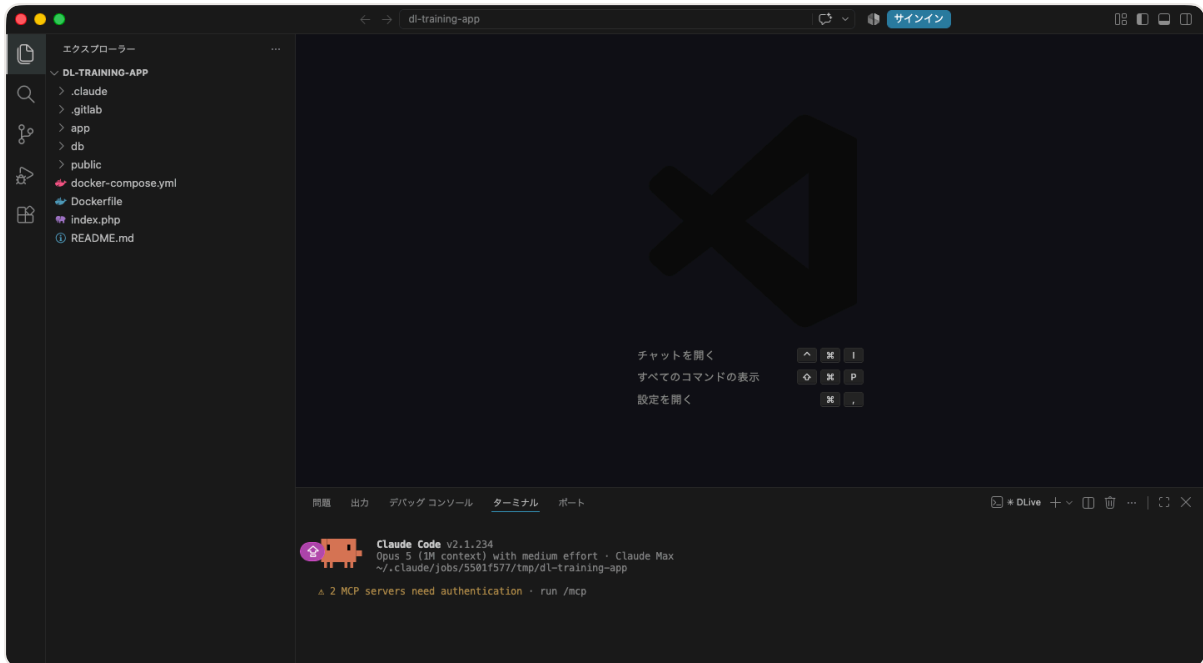
Claude Code は、ターミナルのコマンド（CLI）としても同じことができます。当日の説明と画面は Claude Desktop に合わせています。ターミナル版に慣れている方はそちらで進めても構いませんが、画面や操作の説明は読み替えてください。読み替え方は [Step 3](#) の「ターミナル版で進める方へ」にあります。講師の画面と見比べられるよう、Claude Desktop の導入と Step 3 までの確認は、ターミナル版で進める方も済ませておいてください。

研修のあと、自社で使い込む段階では CLI が要る場面があります。Claude Desktop は対話での利用に絞られていて、非対話実行（`-p`）、エージェントチーム、インラインのコード補完、第三者プロバイダ経由の接続の4つは CLI 側にしかありません。定型作業を CI やバッチから自動で走らせるには、このうち非対話実行が要ります。

CLI を入れる場合は Node.js 18 以降が必要です。未導入であれば nodejs.org の LTS 版を入れてから、お使いのターミナルで次を実行してください。

```
npm install -g @anthropic-ai/claude-code
```

導入後はターミナルを開き直し、`claude --version` で 2.1.280 以降の数字が出ることを確認します。初回のログイン手順は [公式のクイックスタート](#) にあります。



ターミナルから起動した場合の画面。画面は講師のMac環境のもので、パスなどの細かい表示はお使いの環境によって変わります

PowerShell の実行ポリシーの確認 Windows のみ

演習のフックは PowerShell のスクリプト(.ps1)です。会社のグループポリシーで実行ポリシーが固定された端末では、スクリプトが読み込まれず、フックが働きません。スタートメニューから Windows PowerShell を開き、次の1行を実行してください。

```
Get-ExecutionPolicy -List
```

表示された一覧のうち、 MachinePolicy と UserPolicy の2行が Undefined になっていれば問題ありません。ほかの行の値は変えなくて構いません。どちらかが Undefined 以外の場合は、その画面をギブリーの担当者へ研修前にお知らせください。 Set-ExecutionPolicy は実行しないでください。

Mac の方 この確認は要りません。Mac では演習のフックに bash 版(.sh)を使い、その中で jq というコマンドを使います。ターミナルで jq --version を実行し、バージョンが表示されることを確かめてください。表示されない場合は、ギブリーの担当者へ研修前にお知らせください。

この Step が終わった状態

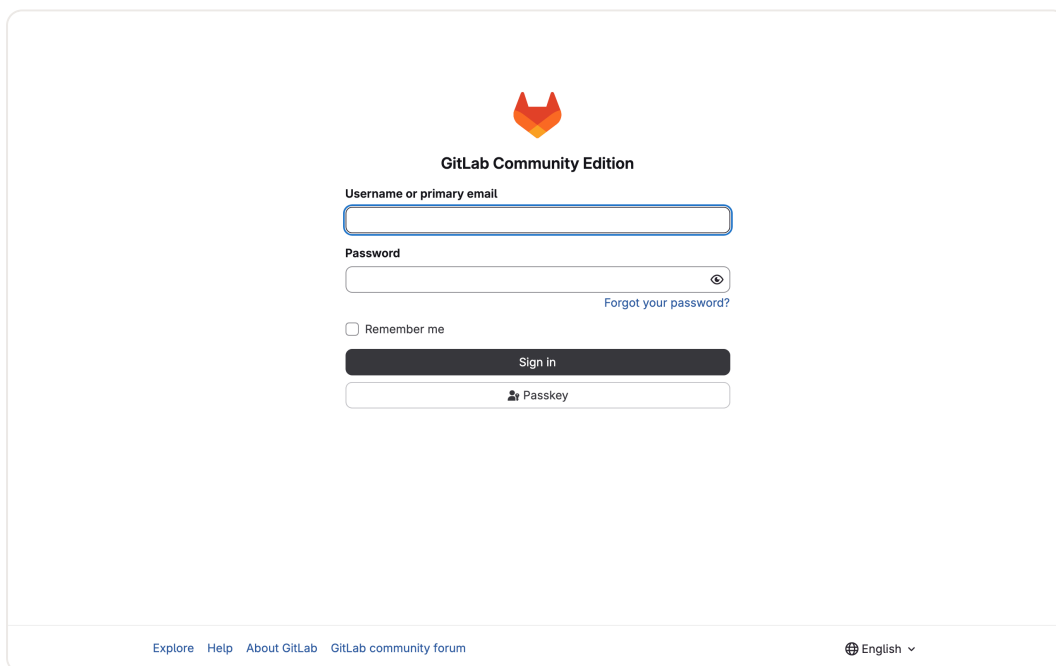
- ☐ Claude Desktop が起動し、会社のアカウントでログインできている
- ☐ VSCode を開ける
- ☐ Windows の方：実行ポリシーの MachinePolicy と UserPolicy が Undefined である。違う場合はギブリーの担当者へ知らせた
- ☐ Mac の方： jq --version でバージョンが表示される

Step 1 GitLab にログインする

STEP 1 配布されたアカウントで GitLab の画面が開く

演習では、変更の共有とレビューを GitLab 上で行います。当日その場でパスワードの初期設定に時間を取られないよう、前日までにログインまで済ませてください。

- 1 ブラウザで <https://gitlab-09291006aidev.give-app.net> を開いてください。下のよう
なサインイン画面が表示されます。
- 2 ユーザー名と初期パスワードでログインしてください。ユーザー名は社用メールアドレスの「@より前」の部分です（例：taro.yamada@……の方はtaro.yamada）。初期パス
ワードは別途ご案内します。
- 3 初回ログインでは、下の「Update password」画面が表示されます。初期パスワード
と、ご自身で決めた新しいパスワードを入力して更新してください。新しいパスワード
は Step 2 でもう一度使いますので、控えておいてください。
- 4 サインイン画面に戻るので、新しいパスワードでもう一度ログインしてください。
- 5 最後に <https://gitlab-09291006aidev.give-app.net/dlive-training/dl-training-app>
を開き、下の画面のようにファイルの一覧が表示されることを確認してください。



The image shows the GitLab Community Edition login interface. At the top is the GitLab logo (an orange fox head) and the text "GitLab Community Edition". Below this are two input fields: "Username or primary email" and "Password". The password field has a toggle icon (an eye) to its right. Below the password field is a link that says "Forgot your password?". There is a checkbox labeled "Remember me" below the password field. At the bottom of the login section are two buttons: a dark grey "Sign in" button and a light grey "Passkey" button with a key icon. At the very bottom of the page, there is a footer with links: "Explore", "Help", "About GitLab", and "GitLab community forum". On the right side of the footer, there is a language selector that says "English" with a dropdown arrow.

手順1と2で表示されるサインイン画面です

Update password for 山田 太郎

To continue, please update your password. After you update, you'll be directed to sign in again.

Signed in as @taro.yamada

Current password

New password

Confirm password

Update passwordSign out

手順3で表示されるパスワード変更画面です。上から順に、初期パスワード、新しいパスワード、新しいパスワード(確認)を入力します

Project

dl-training-app

Pinned

Members

Work items0

Branches

Merge requests0

Pipelines

More features

Manage

Plan

Code

Build

Secure

Deploy

Operate

Monitor

Analyze

Help

Collapse sidebar

dlive-training / dl-training-app

dl-training-app

main dl-training-app

Find fileCode

chore: macOSメタファイル(…)を削除
安田 光憲 authored 1 minute ago

c7fe13cfHistory

Name	Last commit	Last update
.gitlab	chore: macOSメタファイル(…	1 minute ago
app	chore: macOSメタファイル(…	1 minute ago
db	chore: macOSメタファイル(…	1 minute ago
public/css	chore: macOSメタファイル(…	1 minute ago
Dockerfile	dl-training-app v0.1: 演習用…	1 hour ago
README.md	dl-training-app v0.1: 演習用…	1 hour ago
docker-compose.yml	dl-training-app v0.1: 演習用…	1 hour ago
index.php	dl-training-app v0.1: 演習用…	1 hour ago

README.md

ホクトリユース基幹システム(演習用)

Project information

2 Commits

1 Branch

0 Tags

README

Add Wiki

Created on
September 04, 2026

手順5で確認する画面です。app、dbなどのフォルダが見えれば完了です

Tips：ログイン直後のホーム画面で「You haven't contributed to any projects yet.」と表示されますが、異常ではありません。演習プロジェクトはProjects画面の「Member」タブ、または手順5のURLから開けます。

Tips：初期パスワードのご案内が届いていない場合は、ギブリーの担当者へお知らせください。当日の朝でも発行できます。

Tips：このプロジェクトは受講者全員で共有しますが、演習ではお一人ずつ別のブランチで作業するため、他の方の作業には影響しません。ブランチの分け方は当日のハンズオンガイドで

ご案内します。

この Step が終わった状態

- ☐ 初期パスワードから変更が済み、新しいパスワードでログインできる
- ☐ dl-training-app のページを開くと、ファイルの一覧が表示される

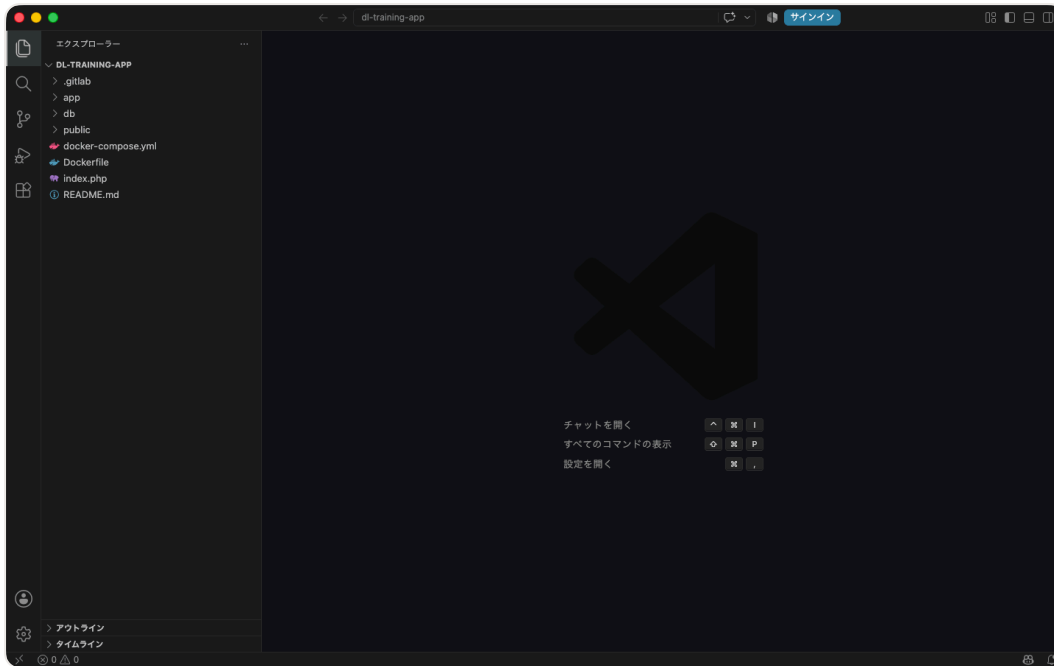
Step 2 演習リポジトリを手元に置く

STEP 2 GitLab の画面から VSCode で開き、演習用のフォルダが手元にある

演習で使う題材アプリを、GitLab の画面から VSCode へ取り込みます。コマンドの入力は要りません。この取り込みは Git が動いていないと進まないで、そのまま Git の導入確認にもなります。普段お使いのエディタや Git クライアント（SourceTree など）がある方は、それで `dl-training-app` を clone しても構いません。演習で VSCode を使うのは、ファイルを読む画面と Git を操作する画面としてで、AI への指示は Claude Desktop から出します。

- 1 Step 1 の手順5で開いた `dl-training-app` のページを、もう一度開いてください。
- 2 ファイル一覧の右上にある **Code** ボタンを押します。開いた一覧の「Open in your IDE」の下から **Visual Studio Code (HTTPS)** を選んでください。
- 3 ブラウザが「Visual Studio Code を開きますか」と確認してくるので、開くを選びます。VSCode が前面に出て、保存先のフォルダを選ぶ画面が開きます。
- 4 保存先のフォルダを選び、画面下の **リポジトリの場所として選択** を押してください。選んだフォルダの下に `dl-training-app` フォルダが作られます。画面の例ではデスクトップを選んでいきます。
- 5 ユーザー名とパスワードを聞かれたら、Step 1 のユーザー名と、変更後のパスワードを入力してください。小さなウィンドウ(Git Credential Manager)が前面に開くことがありますが、入力する内容は同じです。一度入力すると記憶されるので、次回からは聞かれません。記憶先は、Windows では資格情報マネージャー、Mac ではキーチェーンです。
- 6 取り込みが終わると、VSCode が「開きますか」と聞いてきます。開くを選ぶと、左側にファイルの一覧が表示されます。

Tips : Windows の方への注意です。デスクトップが OneDrive の同期対象になっている端末では、`.git` の中身まで同期されて壊れやすくなります。その場合はデスクトップを避け、`C:\work` のように OneDrive の外で、パスに日本語と空白を含まないフォルダを作って保存先にしてください。



手順6まで終わった画面。左側に .gitlab、app、db などのファイル一覧が表示されます

Tips : 「already exists」と出た場合は、前に取得したフォルダが残っています。そのフォルダを VSCode で開いて Step 3 へ進んで構いません。認証で止まったときは、下のよくある質問をご覧ください。

取り込みが進まないとき

手順2で **Visual Studio Code (HTTPS)** を押しても VSCode が前面に出てこない場合や、取り込みの途中でエラーが表示されて止まる場合は、Git が入っていないことが考えられます。社内の標準ソフトウェアの導入手順に沿ってインストールいただくか、ギブリーの担当者へご相談ください。

ソース管理の場所

VSCode の左端にアイコンが縦に並んでいます。枝分かれの形のアイコンが「ソース管理」です。変更したファイルの一覧、コミット、プッシュ、プルは、当日の演習でもすべてこの画面から操作します。いまは場所だけ開いて見ておいてください。使い方は当日のハンズオンガイドでご案内します。

この Step が終わった状態

- ☐ 保存先に選んだフォルダの下に dl-training-app フォルダがある
- ☐ VSCode で開くと、app、db、public などのフォルダが見える
- ☐ 取り込めているので、Git は手元の PC に入っている

Tips : ふだんからターミナルをお使いの方への補足です。上の手順の代わりに、保存先のフォルダへ移動して `git clone https://gitlab-09291006aidev.give-app.net/dlive-training/dl-`

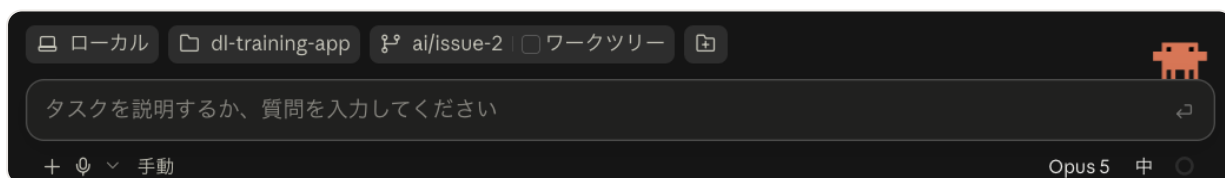
`training-app.git` を実行しても同じ結果になります。研修の進行は画面の操作で統一しますので、どちらでも構いません。

Step 3 Claude Desktop で演習フォルダを開く

STEP 3 Claude Desktop で dl-training-app を開き、応答が返る

当日の演習は、Claude Desktop の中の Claude Code で演習フォルダを開いて進めます。ここでは当日と同じ形で動作確認します。

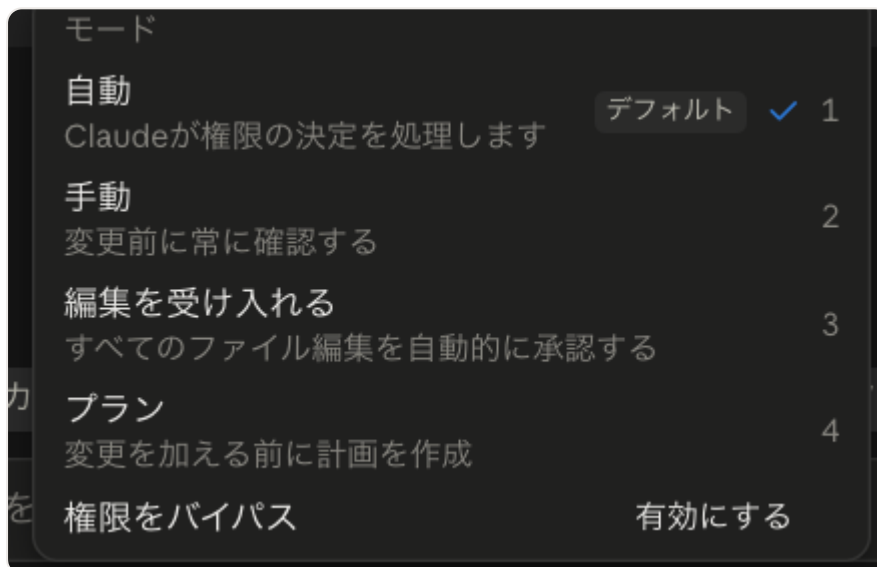
- 1 Claude Desktop の **Code** タブを開きます。最初のメッセージを送る前に、入力欄のまわりで4つを選びます。
- 2 環境は **ローカル**、プロジェクトフォルダは Step 2 で作った dl-training-app、モデルは既定のままで構いません。
- 3 4つ目のパーミッションモードは **手動** を選んでください。当日は Claude からの確認ダイアログを見る演習がありますので、メッセージを送る前にここが **手動** になっていることを確認する癖をつけてください。
- 4 「こんにちは」など何か1つ送り、応答が返ることを確認してください。返事が返り、画面上部の会話名の横に dl-training-app と出ていれば、正しいフォルダで開けています。この画面のまま Step 4 へ進みます。



手順1から3を終えた入力欄です。上の行の左から環境、プロジェクトフォルダ、ブランチ名で、下の行の左がパーミッションモード、右がモデルです。画面は撮影時点のもので、モデル名はいまは Opus 5.5 と出ます



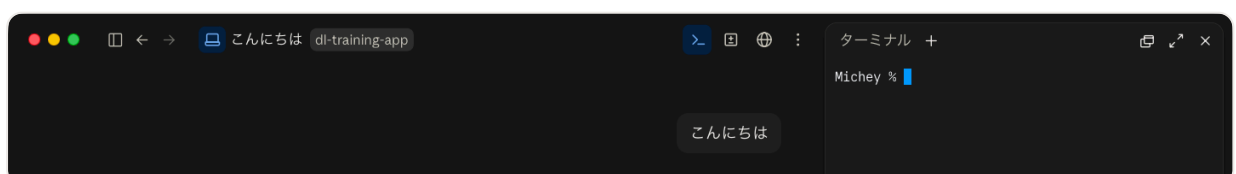
Windows で「こんにちは」と送った直後の画面です。上部の会話名の横に dl-training-app が出ていれば、正しいフォルダで開けています。返事のブランチ名は、そのとき開いているブランチの名前になります



下の行のモード名を押すと開くメニューです。 **手動** を選びます

統合ターミナルの開き方

演習では、変更の差分をまとめてレビューさせるなど、コマンドを打つ場面があります。セッションの右上にあるターミナルのアイコン(), または `Ctrl + バッククォート` (Shift と `@` のキー) で、画面の中にターミナルが開きます。開く場所はそのセッションの作業フォルダで、Claude と同じファイルを見えています。当日のコマンドはすべてここから実行します。次の Step 4 でも、この統合ターミナルを使います。



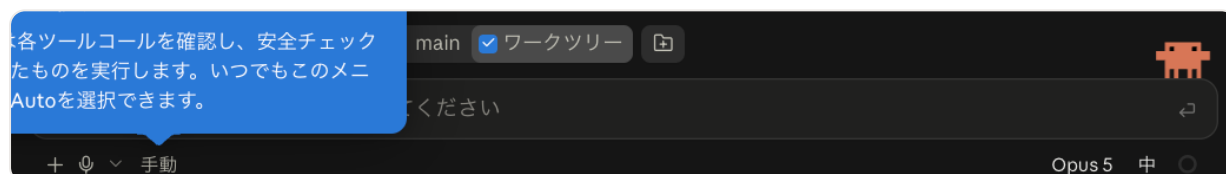
統合ターミナルを開いた画面です。右上のアイコンが押された状態になり、右側にターミナルのペインが出ます。プロンプトが出ていれば、そのまま打てます。画面は Mac で撮ったもので、Windows ではプロンプトが `PS C:\...>` の形になります

WSL の中にあるリポジトリで作業する方へ

Claude Desktop は、セッションを WSL 2 のディストリビューションの中で動かします。セッション開始時の環境の選択で WSL の欄からディストリビューションを選び、`/home/...` の Linux のパスでフォルダを選びます。Claude Code の本体、ツール、git がすべて WSL の中で動くので、ターミナルから WSL の中で起動したときと同じ構成になります。ただし2026年9月時点で、WSL セッションでは統合ターミナルが使いません。コマンドを打つ場面では Windows Terminal で WSL を開き、リポジトリのルートへ移動してから同じコマンドを打ってください。演習リポジトリのフックとテストランナーは bash 版（`.sh`）も入っています。

当日使う worktree

ハンズオンでは、AI に触らせる範囲をブランチごとに分ける worktree を使います。セッションを始めるときにブランチ名の右の **ワークツリー** にチェックを入れると、`.claude/worktrees/` の下に作業用のコピーが作られます。使い終わったセッションは、サイドバーのアーカイブのアイコンで片付けます。`.env` のように Git の管理から外したファイルを持ち込む方法もあわせて当日ご案内しますので、事前の準備は要りません。



ブランチ名の右にある **ワークツリー** のチェックです。当日はここを使う場面で講師が案内します。事前に入れておく必要はありません

ターミナル版で進める方へ

当日の説明と画面は Claude Desktop に合わせています。ターミナル版（`claude` コマンド）に慣れている方はそちらで進めても構いませんが、画面や操作の説明は次のように読み替えてください。演習でやること、置くファイル、フックの効き方はどちらでも同じです。手順の「入力欄に送る」はターミナルの入力に、「統合ターミナルで」は別のターミナルを開いて同じフォルダで打つ、と読み替えてください。パーミッションモードの切り替えは、ターミナル版では起動時の `--permission-mode` で行います。

Tips：設定は Claude Desktop と CLI で共通です。 `~/.claude/settings.json` の `permissions` とフック、プロジェクトの `CLAUDE.md` は、どちらから起動しても同じように効きます。当日つくる設定ファイルも同じです。

Tips：画面のスクリーンショットや PDF は、入力欄に添付してそのまま渡せます。ほかのフォルダでは動くのに `dl-training-app` でだけ応答が返らないときは、その画面をギブリーの担当者へお送りください。当日までにこちらで原因を確認します。

この Step が終わった状態

- ☐ Claude Desktop で `dl-training-app` を開き、応答が返る

☐ パーミッションモードが手動になっている

Step 4 コミットの作者名の設定

STEP 4 名前とメールアドレスを登録し、GitLab に表示される作者名を決める

演習で作る変更には、作った人の名前とメールアドレスが記録されます。これが未設定のままだと、GitLab 上で誰の変更か分からなくなります。Step 3 で開いた Claude Desktop の統合ターミナルから設定します。

1 Step 3 で開いた dl-training-app のセッションを表示します。右上のターミナルのアイコン(>_)、または Ctrl とバッククォート (Shift と @ のキー) で、画面の中にターミナルを開いてください。

2 開いた場所はそのセッションの作業フォルダなので、フォルダの移動は要りません。次の2行をそのまま貼り付けて Enter を押します。氏名と社用メールアドレスはご自身のものに置き換えてください。

```
git config --global user.name "氏名 (ローマ字可)"
git config --global user.email "社用メールアドレス"
```

3 何も表示されずに次の入力待ちに戻れば設定できています。念のため確認したい場合は、Claude の入力欄に「git のユーザー設定を確認して」と送ってください。設定した名前とメールアドレスが返ってきます。

Tips :すでに別の案件で設定済みの方は、そのまま結構です。当日の演習は、いま入っている名前のままで問題なく進みます。

この Step が終わった状態

- ☐ 統合ターミナルを開ける
- ☐ 名前とメールアドレスが登録されている

Step 5 最終確認チェックリスト

STEP 5

開催前日までに、ここが全部そろっていれば準備完了です

確認事項

- ☐ VSCode が開けて、Claude Desktop にログインできている ([Step 0](#))
- ☐ Windows の方は実行ポリシーの `MachinePolicy` と `UserPolicy` が `Undefined` である。Mac の方は `jq --version` でバージョンが表示される ([Step 0](#))
- ☐ GitLab にログインでき、パスワードを変更した ([Step 1](#))
- ☐ dl-training-app フォルダがある。Windows の方は OneDrive の同期対象の外に置いた ([Step 2](#))
- ☐ Claude Desktop で dl-training-app を開くと、応答が返る ([Step 3](#))
- ☐ コミットの作者名（名前とメールアドレス）を設定した ([Step 4](#))

うまくいかない Step があっても、開始前の時間で個別に対応します。どこで止まったかだけメモしておいてください。

よくある質問

会社の端末から GitLab につながりません

社内の通信制限で止まっている可能性があります。エラー画面をそのままギブリーの担当者へお送りください。接続先のドメインをお伝えしますので、社内の解除手続きにお使いください。

Claude Desktop を持っていません

Step 0 の「Claude Desktop の導入」のとおりに入れてください。社内の申請などで導入できない場合は、ギブリーの担当者へ事前にお知らせください。進め方を個別にご相談します。

Docker のインストールは必要ですか

不要です。演習コードの実行とテストは、こちらでご用意した GitLab 上で自動的に行われます。お手元の PC に入れていただくものは、Git・VSCode・Claude Desktop だけです。

演習用のリポジトリに自社のコードを置いてもよいですか

置かないでください。演習は配布する題材だけで完結します。実際のコードや業務データは持ち込まずにお進めください。

ユーザー名とパスワードを何度も聞かれます

パスワードが違うか、以前の間違ったパスワードが Windows に記憶されています。Step 1 で変更した新しいパスワードを使っているかを確認し、それでも続く場合は Windows の「資格情報マネージャー」で `git:https://gitlab-09291006aidev.give-app.net` の行を削除してからやり直してください。Mac の方は「キーチェーンアクセス」で `gitlab-09291006aidev.give-app.net` の項目を削除します。

Claude Desktop が起動しません

一度アプリを終了してから開き直してください。それでも同じ場合は、Step 0 のインストーラをもう一度実行すると上書きで直ることがあります。npm で CLI を導入した方は、ターミナルを閉じて開き直してください。npm が「認識されていません」と出る場合は Node.js が入っていません。

Mac で参加します

手順は同じです。Claude Desktop も VSCode も macOS 版があり、画面の見た目と操作は Windows とほぼ変わりません。演習中に配布するフック類は Windows 版と bash 版の両方を用意しているので、bash 版をお使いください。bash 版は `jq` を使うので、Step 0 の Mac の方向けの確認を済ませておいてください。PowerShell の実行ポリシーの確認は要りません。



AIカスタム研修 / 2026年9月29日・10月6日 / データライブ株式会社様向け

© Givery, Inc. All rights reserved.

※本教材の演習に登場する企業・人物・数値はすべて架空です。