



PRE-SETUP

事前セットアップガイド

本研修では、VSCode に入れた Claude Code から Amazon Bedrock 経由で Claude Sonnet 4.6 を呼び出し、業務アプリを一から作るどころと、Java・Spring Boot のアプリの不具合を直すところを、その場で手を動かしながら進めます。当日いきなり環境構築から始めると、それだけで前半が終わってしまいます。このガイドは、その準備を先に済ませておくためのものです。VSCode を入れたことがない方でも、上から順に進めれば完了します。Windows と Mac のどちらでも受講できます。

🕒 所要 60～90分（回線速度により前後します）

💻 Windows 10 / 11・macOS

📖 15セッション・チェックリスト付き

本ガイドの構成

01 このガイドの読み方

所要時間、進め方、どこまでできれば準備完了かの基準を先に示します。

02 着手前にご確認いただくこと

通信の許可、PCの権限、事務局から受け取る接続情報の3点です。

03 必要なものと想定環境

入れるソフトと、それぞれを研修のどこで使うかの対応表です。

04 全体の流れ

これから何を、どの順番で、どれくらいの時間で行うかを俯瞰します。

05 VSCode のインストール（Windows）

公式サイトを開くところからインストーラーの選択肢まで説明します。

06 VSCode のインストール（Mac）

ZIP の展開先と、初回起動時に出る確認ダイアログへの対応です。

07 VSCode の日本語化

メニュー表示を日本語へ切り替えます。方法は2通りあります。

08 Claude Code 拡張のインストール

本研修の主役です。似た名前の拡張が並ぶため、発行元で見分けます。

09 Amazon Bedrock への接続設定
環境変数を4つ設定します。値は研修事務局が指定します。

10 JDK 17 の準備
課題B の Java アプリを動かすために使います。Temurin を入れます。

11 配布ファイルの展開
展開する場所に条件があります。VSCode で開くフォルダも決まっています。

12 動作確認
Claude Code と1往復し、Java の初回ビルドを済ませておきます。

13 準備完了チェックリスト
研修前日までに、この9項目すべてにチェックが入る状態にします。

14 トラブルシューティング
つまづきやすい14件を、症状から引ける形でまとめています。

15 参考リンクと当日の持ち物
本ガイドで参照した公式ドキュメントと、当日お持ちいただくものです。

SESSION 01

このガイドの読み方

最初に、このガイドを終えたときにどうなっているかを示します。ゴールが見えていれば、途中で「これは何のための作業か」で迷いません。

01 このガイドが目指す状態

ゴール 1

VSCode が日本語で使える

Visual Studio Code（以下 VSCode）を入れて起動し、メニューが「ファイル」「編集」のように日本語で表示される状態です。VSCode はプログラムを書くための無料のアプリで、本研修ではこの中ですべての作業を行います。

ゴール 2

Claude Code が応答を返す

VSCode の左端にある縦のアイコン列（アクティビティバー）に Claude Code のアイコンが出ていて、そこを押して開いたパネルに日本語で話しかけると返事が返ってくる状態です。ここが動けば研修の主要部分は動きます。

ゴール 3

Java のアプリがビルドできる

課題B で使う Java のアプリが、手元でエラーなく組み立て（ビルド）できる状態です。
初回だけ数分かかる処理を事前に済ませておくと、当日の待ち時間が数十秒に縮みます。

02 所要時間の目安

すでに VSCode が入っている方は、SESSION 05・06 を飛ばせます。回線が遅い環境ではダウンロードだけで時間を取られるため、余裕のある日に進めてください。

作業	目安	時間のほとんどを占めるもの
VSCode の導入と日本語化	15分	インストーラーのダウンロード（約 100～200MB）
Claude Code 拡張の導入	5分	拡張機能の取得。数十秒で終わります
Bedrock への接続設定	10分	環境変数4つの入力と、VSCode の再起動
JDK 17 の導入	15分	インストーラーのダウンロード（約 170～190MB）
配布ファイルの展開と動作確認	15～45分	Java の初回ビルド。ライブラリの取得で数分から十数分

Tips 途中で中断しても大丈夫です

各セッションは独立しています。時間が取れないときは SESSION 05 から 09 まで（VSCode と Claude Code）を先に済ませ、JDK と配布ファイルは別の日に回してください。順番を入れ替えても問題ありません。

03 各セッションの読み方

どのセッションも同じ形で書いています。操作の説明のあとに、必ず「これができていればOK」という灰色の枠を置いています。次に進む前に、この枠に書かれた状態になっているかだけ確認してください。

✓ これができているとOK

この枠が「確認方法」です。画面に何が表示されていれば正しいのか、どのコマンドを打って何が返ればよいのかを、毎回ここに書いています。ここが合っていれば、途中の細かい違いは気にしないで構いません。

Windows と Mac で手順が分かれる箇所は、左右2列に並べています。ご自身の環境の列だけを読んでください。折りたたまれた「+」の見出しは補足です。詰まっていなければ開かなくて構いません。

04 うまくいかないとき

まず SESSION 14 のトラブルシューティングを見てください。つまりきやすい14件を、症状から引ける形でまとめています。それでも解決しない場合は、次の3点を添えて研修事務局までご連絡ください。

- どのセッションのどの手順で止まったか（例: SESSION 09 の手順3）
- 画面に出ているメッセージ（スクリーンショットか、文字をそのままコピーしたもの）
- OS の種類（Windows 10 / Windows 11 / macOS のいずれか）

Tips 連絡いただくときの書き方

「動きません」だけでは原因を絞れません。「SESSION 09 の確認方法で `echo` を実行したが何も返ってこない。Windows 11。PowerShell は開き直した」のように、どこまでできて、どこで止まったかを書いていただけると、その場で解決できることが増えます。

注意 当日の朝に慌てないために

会社のネットワークが外部への通信を制限している場合、インストールそのものがないことがあります。これは受講者側では解決できません。SESSION 02 の確認事項を情報システム部門へ先に共有しておく、当日になってから止まる事態を避けられます。

着手前にご確認ください

セットアップを始める前に、環境側の条件を3つだけ確認します。ここが塞がっていると、以降のセッションはどこかで必ず止まります。

01 通信が許可されているか

本研修のセットアップでは、次のサイトから配布物を取得します。社内ネットワークで外部通信が制限されている場合、この一覧を情報システム部門へ共有し、通信の許可を依頼してください。研修当日も、Claude Code が Amazon Bedrock と通信し続けます。

接続先	使う場面	止まると何が起きるか
<code>code.visualstudio.com</code>	SESSION 05・06	VSCode 本体をダウンロードできません
<code>marketplace.visualstudio.com</code> <code>*.vsassets.io</code>	SESSION 07・08	日本語化パックと Claude Code 拡張を導入できません
<code>adoptium.net</code> <code>github.com</code> （配布物の取得元）	SESSION 10	JDK 17 をダウンロードできません
<code>bedrock-runtime.</code> <code><リージョン>.amazonaws.com</code>	SESSION 12・ 研修当日	Claude Code が応答を返しません。研修が成立しません
<code>repo.maven.apache.org</code>	SESSION 12	Java アプリのビルドが依存ライブラリの取得で失敗します

プロキシ経由の場合

社内プロキシを通す環境では、単に通信を許可するだけでなく、VSCode 側にもプロキシの設定が必要ことがあります。設定方法は SESSION 14 のトラブル 08 に書いています。プロキシのアドレスは情報システム部門にご確認ください。

02 PC にソフトを入れられるか

VSCode と JDK 17 をインストールします。会社支給の PC で管理者権限が制限されている場合、インストーラーが途中で止まります。

WINDOWS

User Installer なら権限なしで入ります

VSCode には System Installer と User Installer の 2種類があります。後者はログインしているユーザーのフォルダ配下に入るため、管理者権限を求められません。本ガイドでは User Installer を使います。JDK は管理者権限が要る場合があります。

MAC

アプリケーションフォルダへの書き込み権限

VSCode は「アプリケーション」フォルダへドラッグして入れます。ここへの書き込みが制限されている場合は、ユーザーのホームフォルダ配下に置いても動作します。JDK のインストーラーは管理者パスワードを求めます。

03 事務局から受け取るもの

次の2点は、研修事務局から別途ご案内します。ご自身で用意するものではありません。案内が届いていない場合は事務局へご連絡ください。

受け取るもの	使う場面	内容
Amazon Bedrock の接続情報	SESSION 09	リージョンと認証情報。研修専用発行され、研修終了後に無効化されます
配布ファイル <code>genai-nyumon-handson.zip</code>	SESSION 11	当日手を動かす題材一式。課題Aと課題Bの材料が入っています

注意 接続情報の取り扱い

接続情報は研修専用発行された認証キーです。第三者への共有、公開された場所への貼り付け、AI への入力はしないでください。誤って外部へ出た場合は、使わずに事務局へご連絡ください。

Tips 個人の契約は不要です

本研修は Amazon Bedrock 経由で Claude を呼び出します。Anthropic の個人アカウントや有料プランの契約は要りません。すでにお持ちの方も、本研修では事務局が配布する接続情報を使ってください。

必要なものと想定環境

入れるものは全部で4つです。それぞれが研修のどの場面で必要になるかを合わせて示します。何のために入れるのかがわかっていると、途中でエラーが出たときに切り分けやすくなります。

01 入れるものと使う場面

入れるもの	それは何か	研修のどこで使うか	本ガイドの該当箇所
VSCode	プログラムを書くための無料のアプリ。 Microsoft 製	最初から最後まで。 すべての作業がこの 中で完結します	SESSION 05・06
日本語化パック	VSCode の メニュー表示を 日本語に切り替える 追加部品	研修全体。講師の 画面と表示を 揃えます	SESSION 07
Claude Code 拡張	日本語の指示で ファイルの作成・ 編集・実行まで進める AI ツール	研修全体。本研修の 主役です	SESSION 08・09
JDK 17	Java のプログラムを 動かすための土台。 Eclipse Temurin 版を 使います	課題B のみ。Spring Boot のアプリを 起動します	SESSION 10

Tips Java を書けなくても受講できます

課題B は Java のアプリを題材にしますが、Java の文法を覚えている必要はありません。コードを読み解いて不具合の見当をつける作業は AI に任せ、受講者は「どこがおかしいか」「どう直すべきか」を判断する側に回ります。JDK は、そのアプリを実際に動かして目で確かめるために入れます。

02 想定環境

区分	構成内容
OS	Windows 10 / Windows 11、もしくは macOS（Apple Silicon・Intel いずれも可）
エディタ	Visual Studio Code の最新版（Claude Code 拡張は 1.94.0 以上が必要です）
AI ツール	Claude Code for VS Code（発行元 Anthropic）。接続先は Amazon Bedrock 経由の Claude Sonnet 4.6 です
言語・ランタイム	JDK 17（Eclipse Temurin を推奨）。課題Bの Spring Boot アプリで使います
ビルドツール	Maven。配布プロジェクトに Maven Wrapper（ <code>mvnw</code> / <code>mvnw.cmd</code> ）が同梱されているため、Maven 本体の個別インストールは不要です
ブラウザ	Chrome または Edge。課題Bで作ったアプリの画面を確認します
ハンズオン題材	配布ファイル <code>genai-nyumon-handson.zip</code> （課題A: 業務アプリを一から作る / 課題B: 既存アプリの不具合を直す）
ディスク空き容量	2GB 程度。内訳は VSCode が約 400MB、JDK 17 が約 350MB、Java の依存ライブラリが約 400MB です

Git と GitHub は使いません

本研修では Git や GitHub の操作を一切行いません。配布ファイルを展開し、VSCode で開いて進めます。Git を入れていなくても受講できます。課題Bで作る Issue も、GitHub 上のもではなく手元の Markdown ファイルです。

03 Claude Code とはどういう道具か

先に道具の性質を掴んでおくと、設定の意味がわかります。Claude Code は、質問に答えるだけの AI ではありません。「一覧画面に検索欄を足してください」と書くと、自分で関係するファイルを探して読み、変更内容を決め、ファイルを書き換え、必要ならコマンドを実行するところまで進めます。人間は結果を確認して、次の指示を出す側に回ります。

できること

読む・書く・実行する

フォルダの中を自分で調べ、複数のファイルにまたがる変更を行い、ビルドやテストのコマンドを実行して結果を読み取ります。研修ではこの一連の流れを何度も回します。

前提が要る

渡した情報の範囲でしか判断できない

社内のルールや、そのプロジェクト特有の事情は知りません。何も渡さなければ一般論で書きます。研修の後半では、その前提をファイルで渡す方法を扱います。

確認が要る

もっともらしい間違いを書くことがある

存在しない機能を、あるかのように書くことがあります。研修では、出てきたものをそのまま信じずに確かめる手順を組み込みます。

SOURCES

Visual Studio Code 公式サイト
Use Claude Code in VS Code (Claude Code Docs)
Claude Code on Amazon Bedrock (Claude Code Docs)
Eclipse Temurin JDK 17 リリース一覧 (Adoptium)
Amazon Bedrock (AWS)
Maven Wrapper (Apache Maven)

SESSION 04

全体の流れ

セットアップは7段階です。前の段階が終わっていないと次に進めないため、順番どおりに進めてください。JDK の導入だけは、他と独立しているので前後しても構いません。

01 セットアップの7段階



図1 セットアップの7段階。1から4までは順番どおりに進めます。5の JDK 17 だけは独立しているため、先に済ませても構いません

02 研修当日の流れ

準備が済んでいる前提で組んだ時間配分です。環境構築の時間は取っていません。



図2 当日の時間配分。半分以上が手を動かす時間です。この時間を環境構築に使わずに済むよう、事前準備をお願いします

03 指示がどこを通過して AI に届くか

SESSION 09 で設定する環境変数が何をしているのかは、経路を見るとわかります。Claude Code は本来 Anthropic のサービスに直接つながりますが、環境変数を1つ立てることで、その接続先を Amazon Bedrock へ切り替えます。



図3 指示が通る経路。SESSION 09 で設定する環境変数は、左側の Claude Code 拡張に「右側の Amazon Bedrock を見に行くこと」と、その入口の鍵を教えるためのものです

Tips ファイルは PC の外に出ないのか

出ます。AI に判断させるためには、対象のファイルの中身を送る必要があります。本研修で扱うのは配布した練習用のファイルだけなので問題ありませんが、この性質は業務で使うときの前提になります。当日の座学で、どこまでを渡してよいかの線引きを扱います。

VSCode のインストール (Windows)

Mac をお使いの方は SESSION 06 へ進んでください。すでに VSCode が入っている方は、この節を飛ばして SESSION 07 へ進んで構いません。バージョンが古い場合の更新方法は、この節の最後に書いています。

01 公式サイトを開く

ブラウザで code.visualstudio.com を開きます。検索エンジンで「VSCode ダウンロード」と調べると、公式ではない配布サイトが上位に出ることがあります。必ずこのアドレスから入手してください。

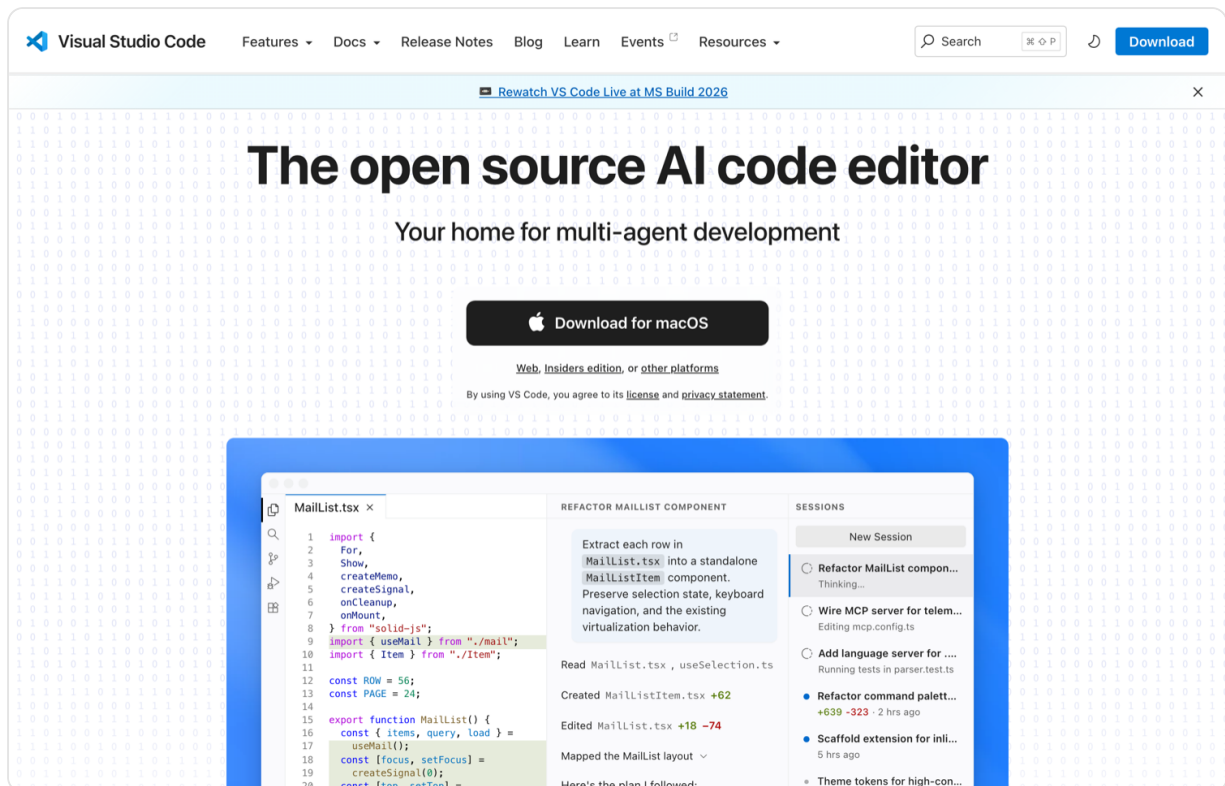


図4 VSCode 公式サイトトップページ。右上の「Download」を押すと、ダウンロードページへ移動します

02 Windows 用のインストーラーを選ぶ

ダウンロードページには Windows・macOS・Linux の3つが並びます。左端の Windows の欄にある「Windows」ボタンを押すと、標準のインストーラーがダウンロードされます。

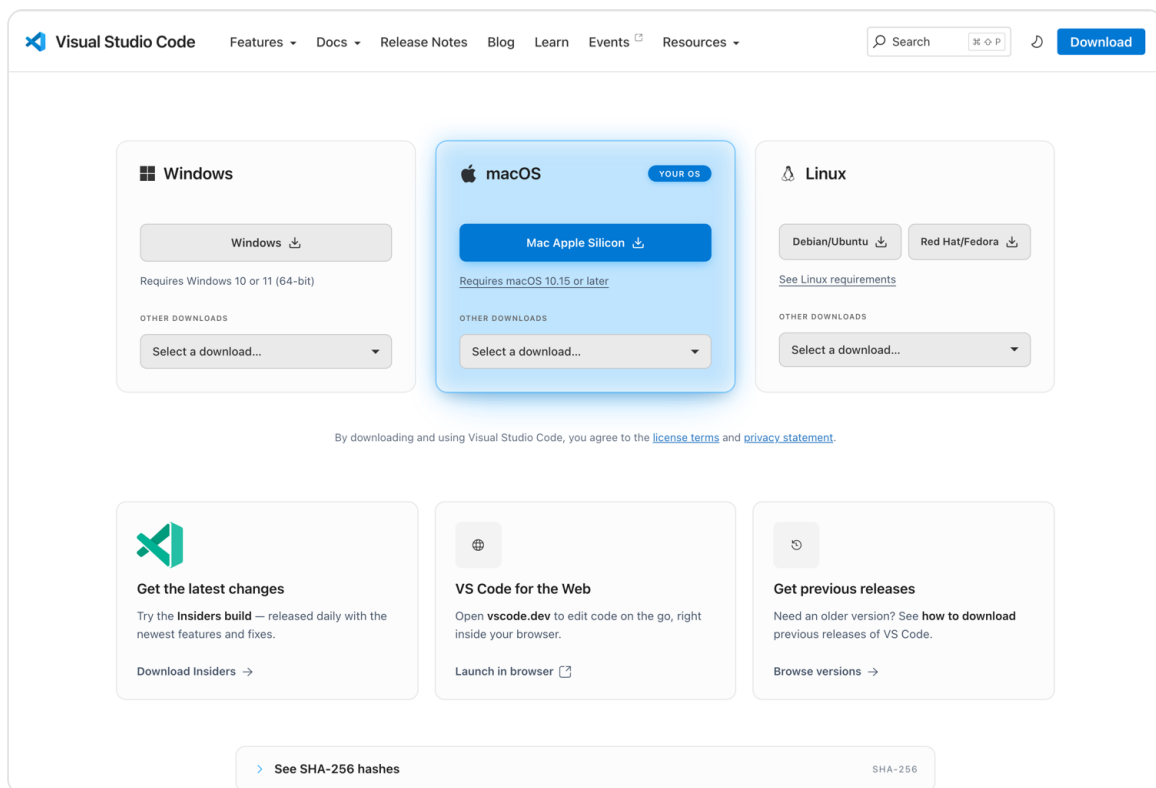


図5 ダウンロードページ。表示中の PC の OS が「YOUR OS」と印されます。Windows の欄には「Requires Windows 10 or 11 (64-bit)」と条件が書かれています

管理者権限を求められて進めない場合

Windows のボタンの下にある「Select a download...」を開くと、インストーラーの種類を選べます。ここから **User Installer** (x64) を選んでください。ログインしているユーザーのフォルダ配下に入るため、管理者権限を求められません。会社支給の PC で権限が制限されている場合は、こちらを使います。

03 インストーラーを実行する

1 ダウンロードしたファイルをダブルクリックします

ファイル名は `VSCodeUserSetup-x64- (バージョン番号) .exe` のような形です。ブラウザの下部か、右上のダウンロード一覧から開けます。

2 使用許諾に同意します

「同意する」を選んで「次へ」を押します。

3 インストール先はそのまま進めます

変更する必要はありません。「次へ」を押し続けます。

4 追加タスクの画面でチェックを確認します

「PATH への追加（再起動後に使用可能）」にチェックが入っていることを確認してください。既定で入っています。ここを外すと、あとでターミナルから VSCode を開けなくなります。他の項目は好みで構いません。

5 「インストール」を押します

1~2分で終わります。完了画面で「Visual Studio Code を実行する」にチェックを入れて「完了」を押すと、そのまま起動します。

スクリーンショット（撮影予定）

Windows のインストーラーの「追加タスクの選択」画面。「PATH への追加」にチェックが入っている状態を枠で囲んで示す

図6 追加タスクの選択画面。PATH への追加が有効になっていることを確認します

04 起動を確認する

スタートメニューから「Visual Studio Code」を探して起動します。「ようこそ」と書かれたタブが表示されれば成功です。この時点では画面はすべて英語です。日本語化は SESSION 07 で行います。

スクリーンショット（撮影予定）

VSCode を初めて起動した直後の Welcome タブ。左端のアクティビティバー、左のエクスプローラー、中央のエディタ領域に番号の吹き出しを付けて名称を示す

図7 起動直後の画面と各部の名称。以降の説明で「アクティビティバー」「エクスプローラー」と書いたときは、ここを指します

✓ これができればOK

VSCode が起動し、「Welcome」または「ようこそ」というタブが開いている。画面の左端に、縦に並んだアイコンの列（アクティビティバー）が見えている。

－すでに VSCode が入っている場合の確認と更新

Claude Code 拡張は VSCode 1.94.0 以上を必要とします。2024年10月より前に入れたきり更新していない場合は、先に更新してください。

1. メニューの「ヘルプ」(Help) から「更新の確認」(Check for Updates...) を選びます
2. 更新がある場合はダウンロードが始まり、再起動を促されます
3. バージョンは「ヘルプ」から「バージョン情報」(About) で確認できます

会社の配布ポリシーで自動更新が止められている場合、この項目自体が出ないことがあります。その場合は情報システム部門へ更新を依頼してください。

VSCode のインストール (Mac)

Windows をお使いの方は SESSION 05 をご覧ください。Mac の場合は、ダウンロードした ZIP を展開してアプリを所定の場所へ移す、という手順になります。ここを飛ばすと、あとで動作が不安定になることがあります。

01 Mac 用のファイルを取得する

ブラウザで code.visualstudio.com を開き、ダウンロードページへ進みます (図5)。macOS の欄に表示されるボタンを押します。

Apple Silicon (M1 以降)

「Mac Apple Silicon」を選びます。
2020年後半以降に発売された Mac のほとんどがこちらです。

Intel 搭載機

「Select a download...」から「Intel chip」を選びます。それより前のモデルはこちらです。

Tips どちらか分からないとき

画面左上のアップルマークから「このマックについて」を開きます。「チップ」の欄に `Apple M1` のように書かれていれば Apple Silicon、「プロセッサ」として `Intel Core` と書かれていれば Intel 搭載機です。判別できない場合、公式サイトは表示中の Mac に合ったものを既定で選ぶため、そのまま押して構いません。

02 アプリケーションフォルダへ移動する

1 ダウンロードした ZIP を展開します

ダブルクリックすると同じ場所に `Visual Studio Code.app` ができます。Safari の場合、ダウンロード時に自動で展開されていることもあります。

2 アプリケーションフォルダへドラッグします

Finder のサイドバーにある「アプリケーション」へドラッグして移動してください。「ダウンロード」フォルダに置いたまま使うと、更新の適用や設定の保存で問題が出る場合があります。

3 ダブルクリックで起動します

初回は「インターネットからダウンロードされたアプリケーションです。開いてもよろしいですか」という確認が出ます。「開く」を押してください。

スクリーンショット（撮影予定）

Finder で Visual Studio Code.app をアプリケーションフォルダへドラッグしている状態と、
初回起動時の確認ダイアログを2枚並べる

図8 アプリケーションフォルダへの移動と、初回起動時の確認ダイアログ

「開発元を検証できないため開けません」と出た場合

macOS のセキュリティ設定によっては、確認ダイアログではなくこの警告が出て開けません。

「システム設定」から「プライバシーとセキュリティ」を開き、画面を下へたどると「"Visual Studio Code" は開発元を確認できないため、使用がブロックされました」という表示と「このまま開く」ボタンが出ます。公式サイトから入手したことを確かめたうえで押してください。詳しくは SESSION 14 のトラブル 02 に書いています。

03 起動を確認する

「Welcome」タブが表示されれば成功です。この時点では画面はすべて英語です。日本語化は次のセッションで行います。

✓ これができればOK

VSCode が起動し、「Welcome」タブが開いている。アプリケーションフォルダに `Visual Studio Code.app` がある（Dock からではなく Finder で確認してください）。

－ ターミナルから VSCode を開けるようにする（任意）

研修中は必須ではありませんが、設定しておくともフォルダを開く操作が速くなります。

1. VSCode を開いた状態で `Cmd + Shift + P` を押します。上部にコマンドパレット（入力欄）が出ます
2. `shell command` と入力し、表示された「Shell Command: Install 'code' command in PATH」を選びます
3. 管理者パスワードを求められたら入力します

以降、ターミナルで `code フォルダ名` と打つと、そのフォルダを VSCode で開けます。

VSCode の日本語化

メニューや設定画面の表示を日本語に切り替えます。研修は日本語表示を前提に進めるため、講師の画面と揃えておくと思いません。方法は2通りあり、どちらでも結果は同じです。うまくいかないときのために両方を載せます。

01 方法A 表示言語の設定から切り替える

公式ドキュメントが案内している方法です。こちらを先に試してください。

1 コマンドパレットを開きます

Windows は `Ctrl + Shift + P`、Mac は `Cmd + Shift + P` です。画面の上部に入力欄が出ます。
ここは VSCode のあらゆる操作を名前呼び出せる場所です。

2 `Configure Display Language` と入力します

途中まで打つと候補に出ます。「Configure Display Language」を選んでください。

3 「日本語」を選びます

言語の一覧が出ます。「日本語 (ja)」を選ぶと、まだ入っていない場合は言語パックのインストールが自動で始まります。

4 再起動します

「Restart」を押すか、VSCode をいったん閉じて開き直します。再起動しないと表示は変わりません。

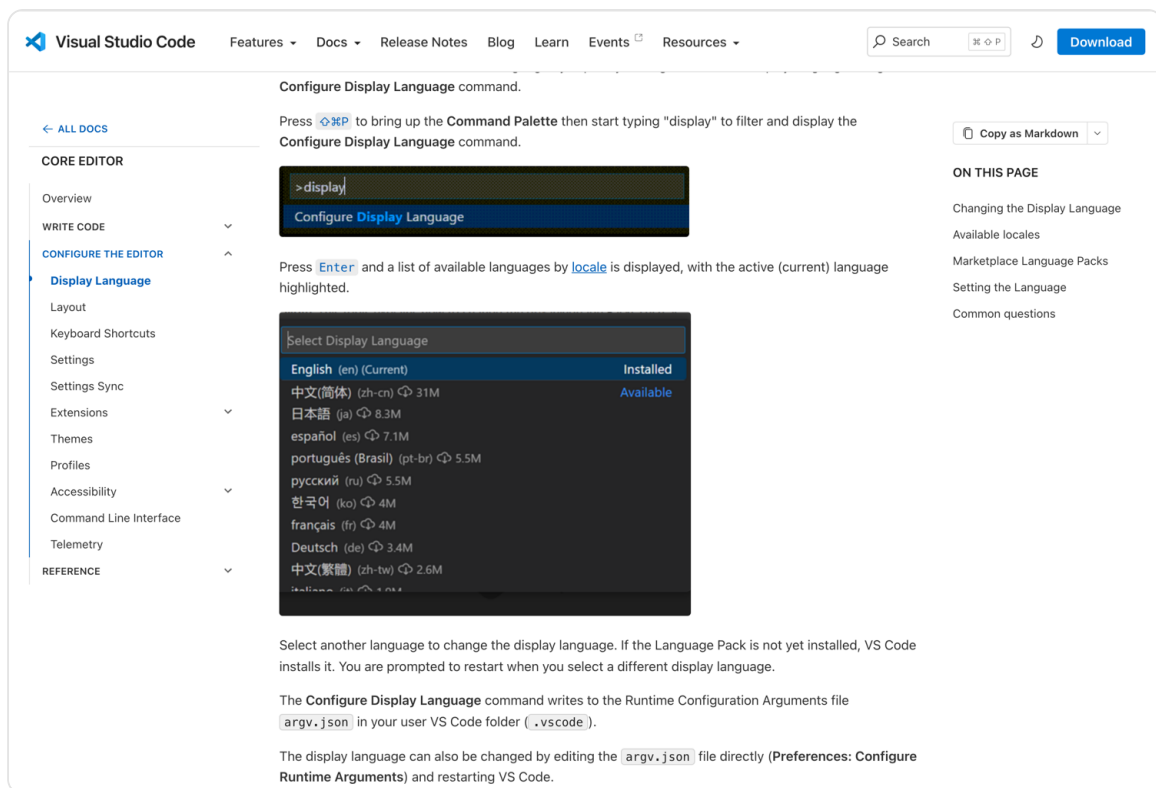


図9 公式ドキュメントの表示言語のページ。コマンドパレットから「Configure Display Language」を実行する手順が案内されています

02 方法B 拡張機能の一覧から入れる

方法A で候補に「日本語」が出てこない場合は、こちらで直接入れます。

1 拡張機能の一覧を開きます

左端のアクティビティバーにある、四角が4つ並んだアイコンを押します。キーボードなら Windows は `Ctrl + Shift + X`、Mac は `Cmd + Shift + X` です。

2 検索欄に「Japanese Language Pack」と入力します

候補が並びます。発行元が **Microsoft** のものを選んでください。名前が似た非公式のものが混ざることがあります。

3 「Install」を押します

数秒で終わります。完了すると右下に「Change Language and Restart」というボタンが出ます。

4 「Change Language and Restart」を押します

VSCode が再起動し、表示が日本語に変わります。ボタンを見逃した場合は、方法A の手順で言語を選び直してください。

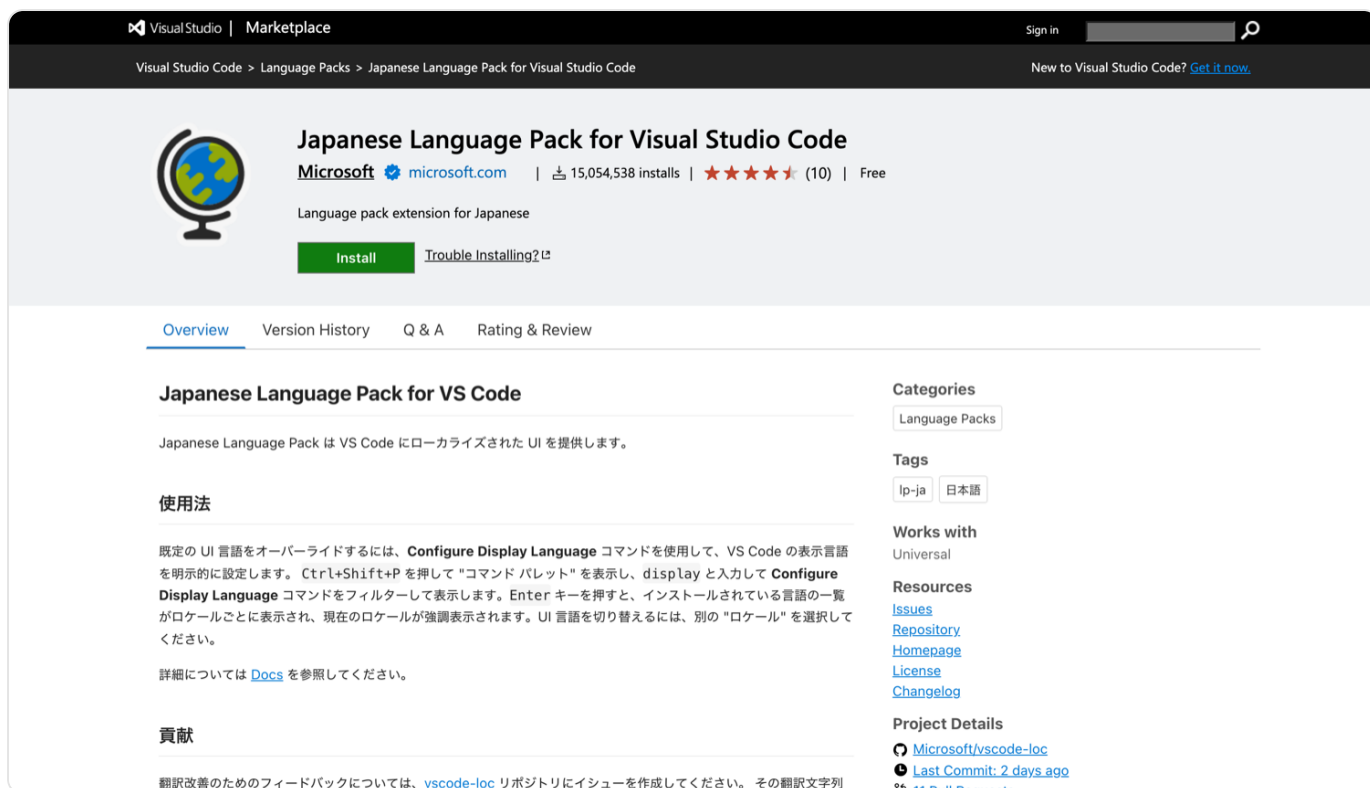


図10 日本語化パックの配布ページ。発行元が Microsoft であること、識別子が `MS-CEINTL.vscode-language-pack-ja` であることを確認できます

スクリーンショット（撮影予定）

VSCode の拡張機能ビューで Japanese Language Pack を検索した結果。発行元 Microsoft の表示と Install ボタンを枠で囲んで示す

図11 拡張機能ビューでの検索結果。発行元の表示を必ず確認します

03 切り替わったか確認する

スクリーンショット（撮影予定）

日本語化後の VSCode。上部メニューが「ファイル」「編集」「選択」「表示」と表示されている状態

図12 日本語化後のメニュー表示

✓ これができているとOK

画面上部のメニューが「ファイル」「編集」「選択」「表示」のように日本語で表示されている。Mac の場合、メニューバーは画面いちばん上に出ます。

まだ英語のままなら、再起動していない可能性が高いです。VSCode を完全に終了してから開き直してください。Mac では、ウインドウを閉じただけではアプリが終了しません。 `Cmd + Q` で終了してください。

Claude Code 拡張のインストール

本研修の主役です。日本語の文章で指示すると、ファイルを探して読み、書き換え、コマンドを実行するところまで自分で進めます。VSCode の拡張として入れて、画面の中で使います。

01 拡張機能の一覧から入れる

1 拡張機能の一覧を開きます

左端のアクティビティバーで四角が4つ並んだアイコンを押すか、Windows は  +  + 、Mac は  +  +  を押します。

2 検索欄に と入力します

名前に Claude を含む拡張が複数並びます。次の手順で見分けてください。

3 発行元が Anthropic のものを選びます

正しいものは、名前が **Claude Code for VS Code**、発行元が **Anthropic** です。名前だけで選ぶと、他の開発者が作った別の拡張を入れてしまうことがあります。

4 「インストール」を押します

数十秒で終わります。完了すると、左端のアクティビティバーに Claude Code のアイコンが追加されます。

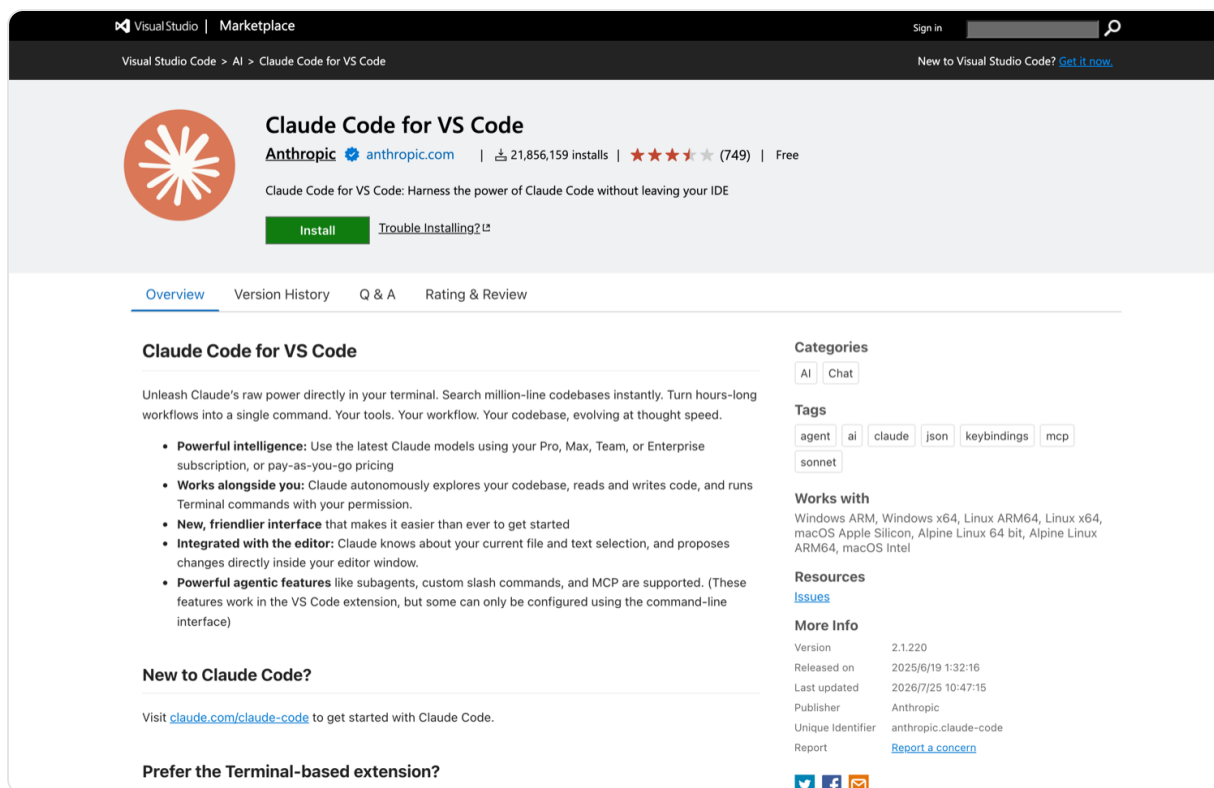


図13 Claude Code 拡張の配布ページ。発行元 **Anthropic**、識別子 `anthropic.claude-code` が目印です。これと違うものは選ばないでください

注意 この時点ではまだサインインしません

インストール直後にサインインを促す表示が出ることがありますが、押さずに閉じてください。本研修は個人アカウントではなく Amazon Bedrock 経由で接続します。次の SESSION 09 で環境変数を設定してから使い始めます。誤ってサインインしてしまった場合も、環境変数の設定が優先されるため、そのまま進めて構いません。

02 アイコンの場所を覚える

研修中は、この拡張のパネルを開いたり閉じたりを繰り返します。場所を先に覚えておいてください。

スクリーンショット（撮影予定）

VSCode の左端アクティビティバーに追加された Claude Code のアイコンを拡大表示し、押した状態のパネル（入力欄が出ている状態）を並べる

図14 アクティビティバーの Claude Code アイコンと、押したあとのパネル

Tips ターミナルからも起動できます

VSCode のターミナルで `claude` と入力しても起動する方法もありますが、こちらは別途コマンドライン版の導入が必要です。研修中はアクティビティバーのアイコンから開く方法で統一します。ご自身の環境では、慣れたほうをお使いください。

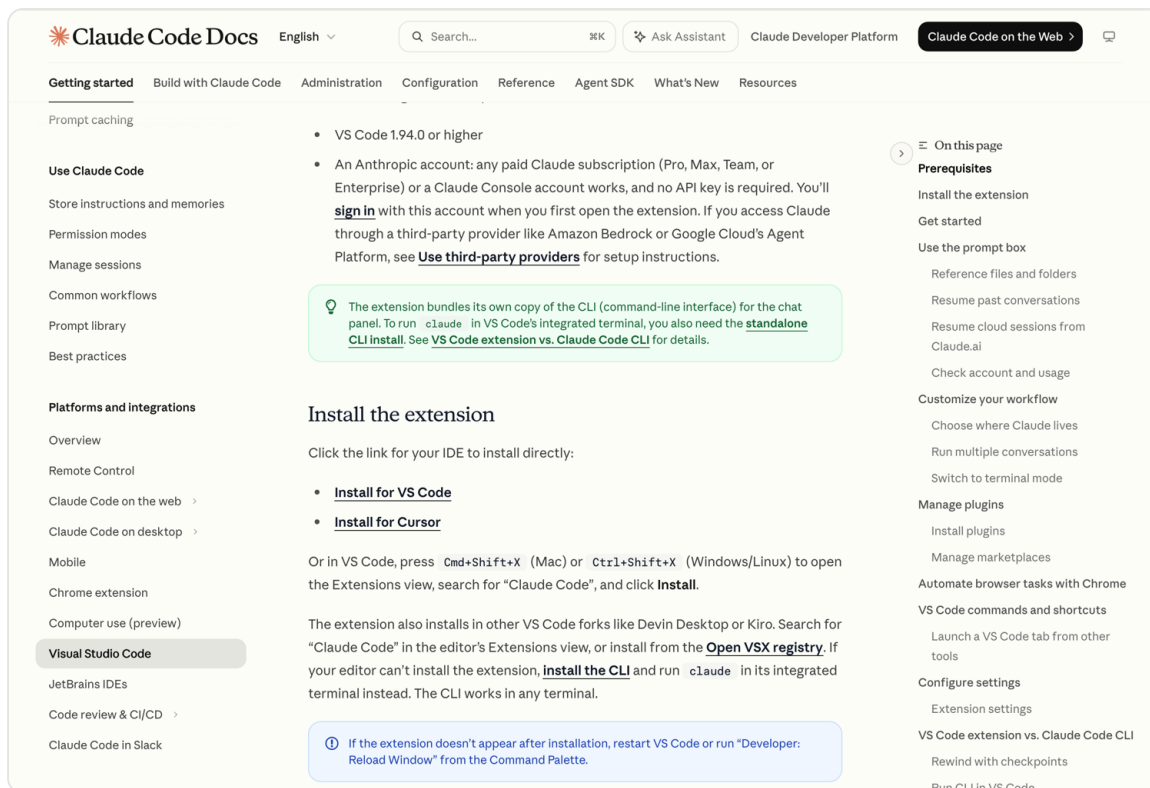


図15 公式ドキュメントの導入手順。VS Code 1.94.0 以上が必要であること、拡張がコマンドライン版を内蔵していることが書かれています

✓ これができればOK

拡張機能の一覧で「Claude Code for VS Code」に「インストール済み」または歯車のアイコンが表示されている。左端のアクティビティバーに Claude Code のアイコンが増えている。

この時点ではまだ応答は返りません。接続先の設定が済んでいないためです。応答の確認は SESSION 12 で行います。

アイコンが見当たらない場合は、SESSION 14 のトラブル 05 に対処を書いています。表示されているアイコンの数が多いと、下部の「…」の中に隠れていることがあります。

Amazon Bedrock への接続設定

Claude Code の接続先を Amazon Bedrock に切り替えます。やることは、環境変数を4つ設定して VSCode を再起動するだけです。設定する値は研修事務局が指定します。案内が届いていない場合は、この節を保留にして SESSION 10 へ進んでください。

01 環境変数とは何か

環境変数は、OS に登録しておく名前付きの設定値です。アプリは起動するときにこれを読み取り、自分の動きを変えます。Claude Code の場合、`CLAUDE_CODE_USE_BEDROCK` という名前の値が `1` になっていると、接続先を Anthropic のサービスから Amazon Bedrock へ切り替えます。残りの3つは、その Bedrock に入るための場所と鍵です（図3 を参照してください）。

環境変数の名前	設定する値	役割
<code>CLAUDE_CODE_USE_BEDROCK</code>	<code>1</code>	接続先を Amazon Bedrock に切り替えるスイッチです。この値だけは全員共通で <code>1</code> です
<code>AWS_REGION</code>	研修事務局が指定します	どの地域の Bedrock を使うかの指定です。 <code>us-west-2</code> のような形の文字列が入ります
<code>AWS_ACCESS_KEY_ID</code>	研修事務局が指定します	AWS に入るための利用者の識別子です
<code>AWS_SECRET_ACCESS_KEY</code>	研修事務局が指定します	上の識別子と対になる秘密の鍵です

配布された案内に5つ目の項目があった場合

一時的な認証情報を配布する場合、`AWS_SESSION_TOKEN` という項目が加わります。案内に載っていれば、下の手順に1行足す形で同じように設定してください。載っていなければ設定は不要です。使うモデルの指定も研修事務局が行います。受講者側で追加の設定をする必要はありません。

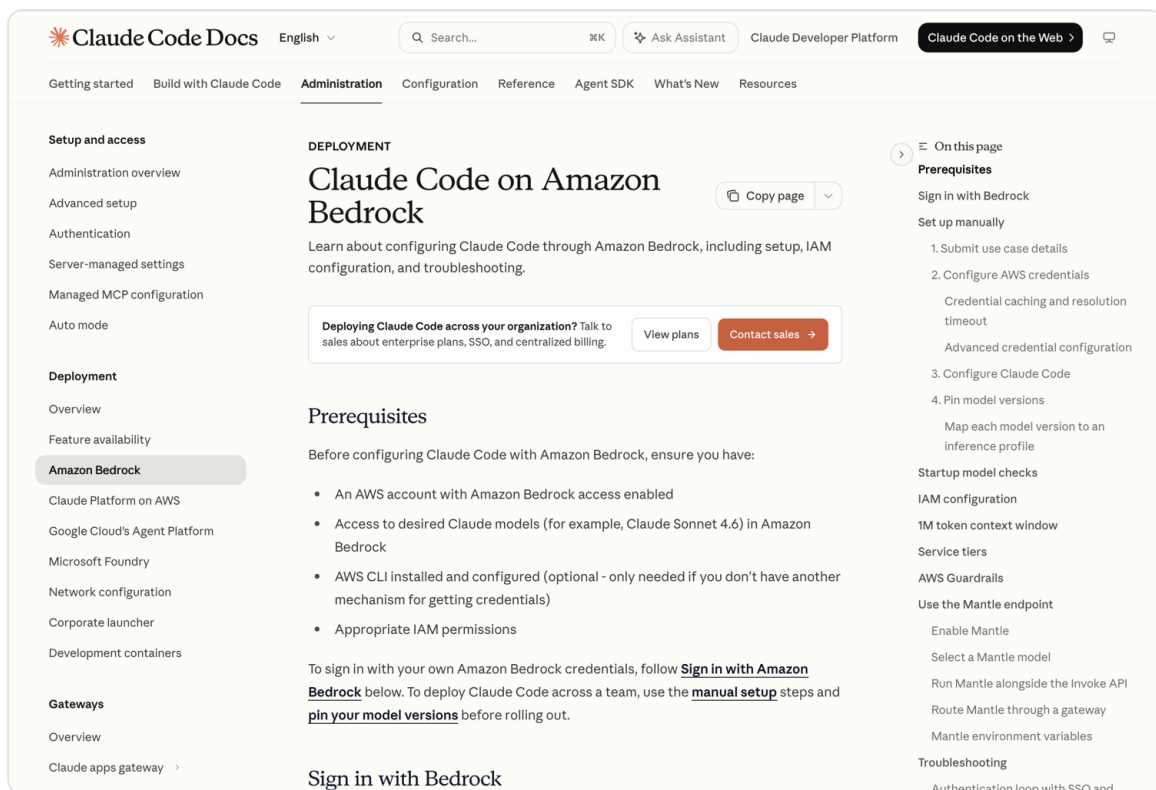


図16 公式ドキュメントのAmazon Bedrock 接続のページ。前提条件として、Bedrock 側で対象のモデルが使える状態になっている必要があると書かれています。この準備は研修事務局が済ませます

02 設定する（Windows）

スタートメニューで「PowerShell」と検索し、**Windows PowerShell** を起動します。次のコマンドを1行ずつ実行してください。 `setx` は、環境変数を次回以降も残る形で登録するコマンドです。

実行する前に、**配布された値** の部分を、案内に書かれた実際の値に置き換えてください。引用符（`"`）は消さずに残します。

```
setx CLAUDE_CODE_USE_BEDROCK 1
setx AWS_REGION "配布された値"
setx AWS_ACCESS_KEY_ID "配布された値"
setx AWS_SECRET_ACCESS_KEY "配布された値"
```

1行ごとに 成功：指定した値は保存されました。 と表示されれば登録できています。

注意 登録しただけでは反映されません

`setx` で登録した値は、そのあとに新しく起動したアプリにだけ反映されます。すでに開いているVSCodeやPowerShellには届きません。**VSCodeを完全に終了してから起動し直してください。**タスクバーに残っている場合は、右クリックして「ウィンドウを閉じる」で終了します。

確認方法

いったんPowerShellも閉じて、新しく開き直してから次を実行します。

```
echo $env:CLAUDE_CODE_USE_BEDROCK
echo $env:AWS_REGION
```

1行目に `1`、2行目に配布された地域の文字列が表示されれば設定できています。何も表示されない場合は、PowerShell を開き直したか確認してください。

03 設定する (Mac)

「アプリケーション」から「ユーティリティ」を開き、「ターミナル」を起動します。次のコマンドを実行すると、ターミナルを開くたびに読み込まれる設定ファイル（`~/.zshrc`）に4行を追記します。

この操作はコピーして貼り付け、まとめて実行して構いません。ただし **配布された値** は先に書き換えておいてください。

```
cat >> ~/.zshrc << 'EOF'
export CLAUDE_CODE_USE_BEDROCK=1
export AWS_REGION="配布された値"
export AWS_ACCESS_KEY_ID="配布された値"
export AWS_SECRET_ACCESS_KEY="配布された値"
EOF
```

Tips このコマンドが何をしているか

`cat >> ファイル名` は、ファイルの末尾に文字を書き足す命令です。`<< 'EOF'` から `EOF` の行までが、書き足す中身です。既存の設定は消えません。最後の `EOF` は行の先頭に置き、後ろに空白を入れないでください。

注意 こちらも再起動が必要です

追記した内容は、そのあとに新しく開いたターミナルから有効になります。VSCode を `Cmd + Q` で完全に終了してから起動し直してください。ウィンドウを閉じただけでは終了していません。

確認方法

ターミナルも新しく開き直してから次を実行します。

```
echo $CLAUDE_CODE_USE_BEDROCK
echo $AWS_REGION
```

1行目に `1`、2行目に配布された地域の文字列が表示されれば設定できています。

Mac のターミナルで echo コマンドを実行し、1 と地域名が返っている状態。認証キーの値は写さない

図17 設定できているときの確認結果。認証キーそのものは表示させないでください

04 設定時によくある取りこぼし

よくある 1

値の前後に空白が混ざる

案内の文面からコピーしたときに、末尾の空白や改行が一緒に入ることがあります。見た目では分かりません。うまくいかないときは、一度手で打ち直してください。

よくある 2

引用符を消してしまう

値に記号が含まれることがあるため、引用符は残したまま中身だけを差し替えてください。" は半角です。全角の ” になっていると失敗します。

よくある 3

再起動していない

接続できない原因のほとんどがこれです。VSCode を完全に終了してから開き直したか、もう一度確認してください。

✓ これができればOK

新しく開いたターミナル（Windows は PowerShell）で `echo` を実行すると、`CLAUDE_CODE_USE_BEDROCK` が `1`、`AWS_REGION` が配布された地域の文字列を返す。そのうえで VSCode を再起動済みである。

Claude Code が実際に応答を返すかどうかの確認は、配布ファイルを展開したあとの SESSION 12 で行います。

SOURCES

Claude Code on Amazon Bedrock（Claude Code Docs）
Claude Code settings と環境変数一覧（Claude Code Docs）
AWS SDK が参照する環境変数（AWS SDKs and Tools Reference）
Amazon Bedrock 製品ページ（AWS）

JDK 17 の準備

課題BでJava・Spring Bootのアプリを動かすために使います。JDKはJavaのプログラムを実行するための土台で、いくつかの提供元があります。本研修ではEclipse Temurinを使います。無料で、WindowsとMacの両方に同じ手順で入れられるためです。

01 先に、入っているか確認する

PCによっては、別の研修や業務ですでにJavaが入っていることがあります。先に確認してください。すでに17が入っていれば、このセッションは飛ばせます。

Windows

スタートメニューで「PowerShell」と検索して起動し、次を実行します。

```
java -version
```

Mac

「アプリケーション」から「ユーティリティ」を開き、「ターミナル」を起動して次を実行します。

```
java -version
```

出力の1行目が `openjdk version "17.0.x"` のように **17** から始まっていると、そのまま使えます。次のSESSION 11へ進んでください。

表示された内容	どうするか
<code>openjdk version "17.0.x"</code>	そのまま使えます。この節は飛ばして構いません
<code>openjdk version "21.0.x"</code> など 17 より新しい	下の手順で 17 を追加で入れてください。新しい版が入っていても、本研修のアプリは 17 で動かします
<code>java version "1.8.0_xxx"</code> など 17 より古い	下の手順で 17 を入れてください
「コマンドが見つかりません」 「認識されていません」	Javaが入っていないか、場所が登録されていません。下の手順で入れてください

02 Temurin の JDK 17 を入れる

ブラウザで adoptium.net の Temurin リリース一覧を開きます。上部のタブで **JDK 17 - LTS** が選ばれていることを確認してください。LTSは長期サポート版という意味で、企業で広く使われている版です。

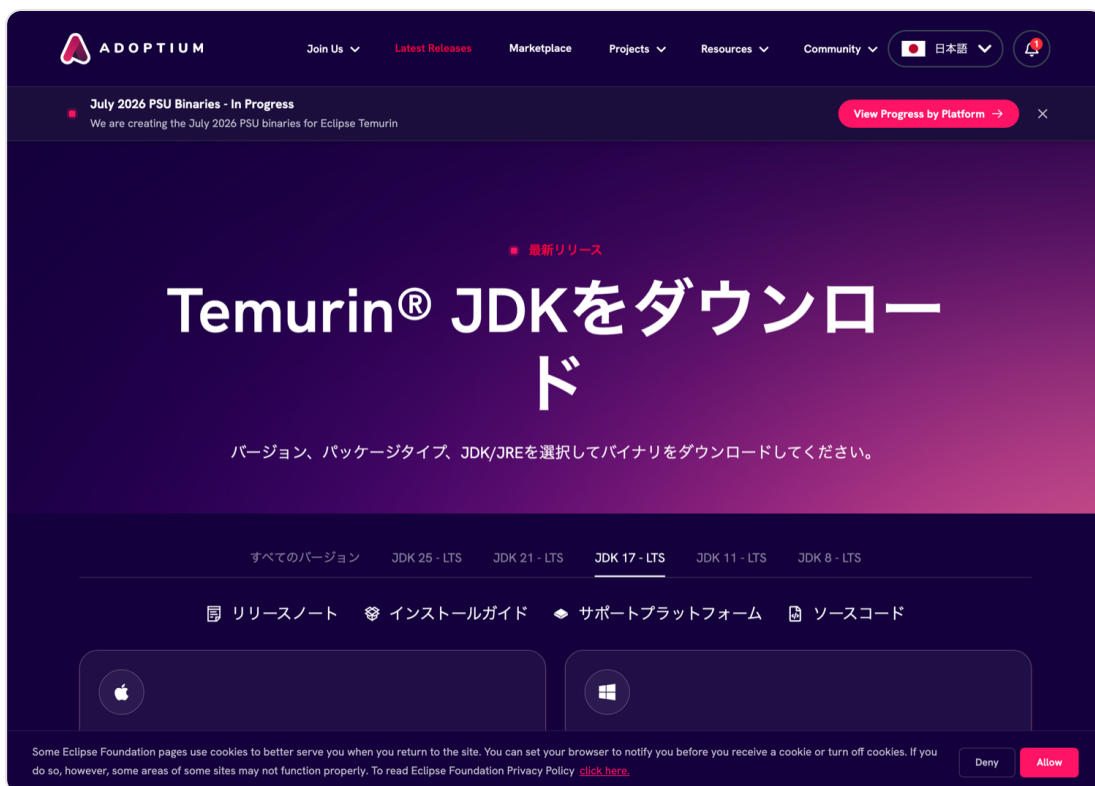


図18 Temurin のリリース一覧。上部のタブで JDK 17 - LTS を選び、下へたどると OS ごとの配布物が並びます

下へたどると、OS ごとの区画が現れます。ご自身の OS の区画から、次のファイルを選んでください。左側の JDK / JRE の切り替えは **JDK** のままにします。JRE は実行専用で、ビルドができません。

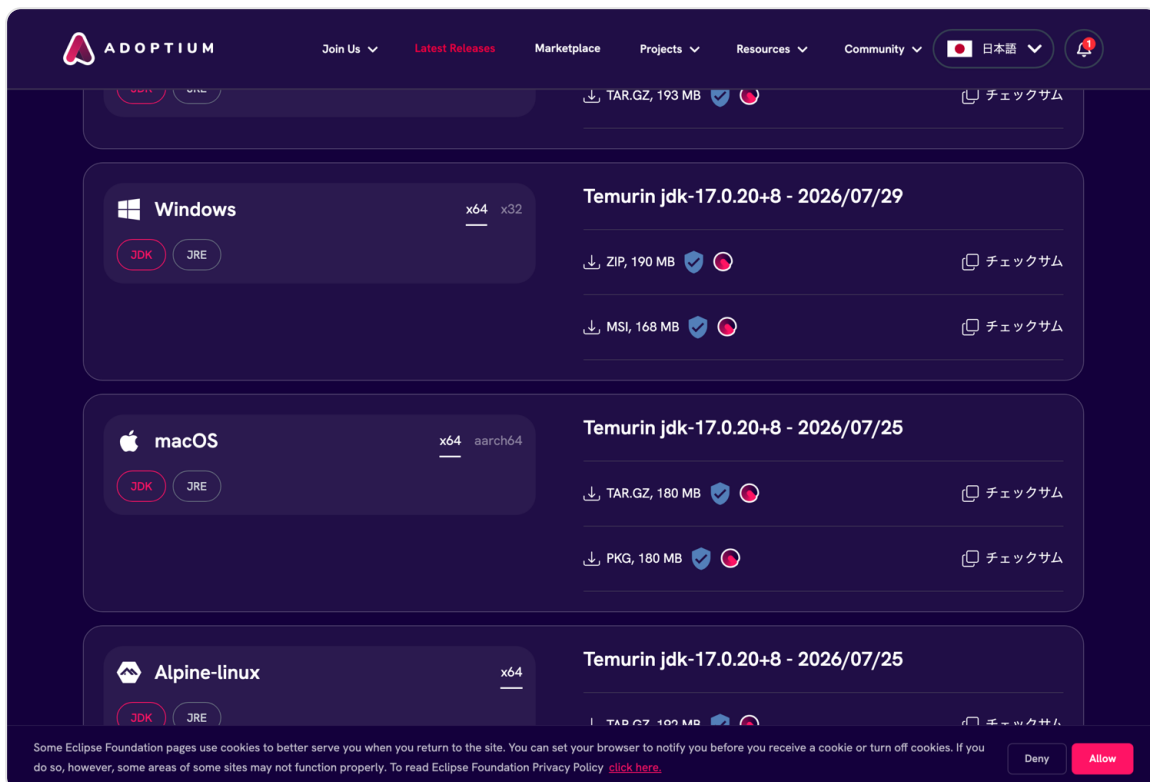


図19 OS ごとの配布物。Windows は MSI、macOS は PKG を選ぶと、インストーラー形式で導入できます

Windows

Windows の区画で **x64** が選ばれていることを確認し、**MSI** をダウンロードします。

実行すると設定画面が出ます。途中の

「カスタムセットアップ」で `Set JAVA_HOME variable` と `Add to PATH` の項目があれば、両方を有効にしてください。既定では無効になっていることがあります。ここを有効にしないと、あとで `java` コマンドが見つからなくなります。

Mac

macOS の区画で、Apple Silicon の場合は **aarch64**、Intel の場合は **x64** を選び、**PKG** をダウンロードします。

ダブルクリックして案内どおりに進めます。途中で管理者パスワードを求められます。追加の設定項目はありません。PATH の設定は自動で行われます。

03 入ったか確認する

インストールが終わったら、**ターミナル (PowerShell)** をいったん閉じて開き直してから、もう一度実行します。開いたままだと、古い状態のままで動くため反映されません。

```
java -version
```

次のような出力になれば成功です。バージョンの細かい数字は時期によって変わります。

```
openjdk version "17.0.20" 2026-07-15
OpenJDK Runtime Environment Temurin-17.0.20+8 (build 17.0.20+8)
OpenJDK 64-Bit Server VM Temurin-17.0.20+8 (build 17.0.20+8, mixed mode)
```

スクリーンショット（撮影予定）

Windows の PowerShell と Mac のターミナルで `java -version` を実行し、Temurin の 17 が表示されている状態を2枚並べる

図20 正しく入っているときの出力。1行目が 17 から始まり、2行目以降に Temurin の名前が出ます

✓ これができているらOK

新しく開いたターミナルで `java -version` を実行すると、1行目が `openjdk version "17.` から始まる。

17 以外が表示される場合は、複数の Java が入っていて別のほうが先にみついています。SESSION 14 のトラブル 10 に切り替え方を書いています。

— そもそも JDK とは何か

Java で書かれたプログラムは、そのままでは PC の上で動きません。人が読める文章から、機械が実行できる形へ変換する道具と、変換したものを動かす土台の両方が要ります。この2つをまとめたものが JDK です。

提供元は複数あり、Oracle 製のものは商用利用の条件が版によって変わります。Eclipse Temurin は Eclipse Foundation が配布している版で、条件を気にせず使えます。企業の開発現場でもよく使われています。

本研修では Java のコードを書きませんが、配布するアプリを実際に動かして画面で確かめるために必要です。

SOURCES

[Temurin JDK 17 リリース一覧 \(Adoptium\)](#)

[Temurin インストールガイド \(Adoptium\)](#)

[Temurin のサポート期間 \(Adoptium\)](#)

配布ファイルの展開

研修事務局から配布される `genai-nyumon-handson.zip` を展開します。展開する場所には条件があります。ここを外すと、当日の Java のビルドで原因の分かりにくいエラーが出ます。

01 展開する場所を決める

次の2つを満たす場所に展開してください。

条件 1

パスに日本語を含まない

「デスクトップ」や「ダウンロード」は、日本語表示でも内部の名前は英語なので問題ありません。避けたいのは「研修資料」「案件」のように、自分で作った日本語名のフォルダの下に置くことです。Java のビルドツールが日本語のパスを扱えず、文字化けやエラーの原因になります。

条件 2

パスに空白を含まない

フォルダ名に空白が入っていると、コマンドが途中で切れて解釈されることがあります。空白の代わりに `_` を使ってください。

迷ったら、次の場所をそのまま使ってください。

```
# Windows
C:\training\

# Mac
/Users/ユーザー名/training/
```

注意 クラウド同期フォルダは避けてください

OneDrive や iCloud Drive、Dropbox の中に置くと、ビルド中に生成される大量のファイルを同期しようとして処理が極端に遅くなったり、書き込みが競合して失敗したりします。同期の対象外の場所に置いてください。Windows の「ドキュメント」フォルダは OneDrive の同期対象になっていることがあるため、注意が必要です。

02 展開する

Windows

1. `genai-nyumon-handson.zip` を右クリックし、「すべて展開」を選びます
2. 展開先に `C:\training` と入力します
3. 「完了時に展開されたファイルを表示する」にチェックを入れて「展開」を押します

ZIP を開いた中身をドラッグして取り出す方法は避けてください。ファイルの一部が取り出されないことがあります。

Mac

1. あらかじめ Finder で `training` という名前のフォルダをホームフォルダに作ります
2. `genai-nyumon-handson.zip` をそのフォルダへ移動します
3. ダブルクリックすると、同じ場所に展開されます

ダウンロードフォルダのまま展開すると、あとで場所が分からなくなりがちです。先に移動しておくのが確実です。

03 展開後の構造と、VSCode で開く場所

展開すると `genai-nyumon-handson` というフォルダができ、その中に `はじめにお読みください.txt`、`資料` フォルダ、`handson` フォルダの3つが入っています。**VSCode で開くのは、この `handson` フォルダです。** 研修中も、最初から最後までここを開いたままにします。`資料` フォルダには本ガイドと手元ガイド共通編の PDF が入っているので、当日は印刷版としても使えます。

genai-nyumon-handson/	
├─ はじめにお読みください.txt	配布フォルダの案内。展開したら最初に読みます
├─ 資料/	本ガイドと手元ガイド共通編の PDF 2本
└─ handson/	← VSCode で開くのはここです
├─ README.md	当日の流れと決めごと。最初に読みます
├─ memo.md	気づいたことを書き残すファイル。見出しだけ入っています
├─ exercises/	演習の手順書。当日ここを見ながら進めます
├─ hints/	詰まったときの参考プロンプト
├─ kadaiA/	課題Aの作業場所。CSV とお題シートだけが入っています
├─ kadaiB/	課題B の Java アプリ。ここでビルドを試します
└─ _harness_kit/	当日、講師の合図があるまで開きません

kadaiA や kadaiB を単体で開くと、演習の途中で配置する設定ファイルが読み込まれません

図21 展開後のフォルダ構造。開くのは `handson` です。ひとつ上の `genai-nyumon-handson` でも、ひとつ下の `kadaiA` でもありません

1 VSCode を起動します

2 「ファイル」から「フォルダーを開く」を選びます

Mac では「フォルダを開く」と表示されます。

3 **handson** フォルダを選んで開きます

フォルダの中に入った状態で「フォルダーの選択」を押してください。ひとつ上の階層を選ばないように注意してください。

4 「このフォルダー内のファイルの作成者を信頼しますか」に答えます

「はい、作成者を信頼します」を選んでください。「いいえ」を選ぶと制限モードになり、Claude Code がファイルを書き換えられません。

スクリーンショット（撮影予定）

VSCode で handson フォルダを開いた状態。左のエクスプローラーに `_harness_kit / exercises / hints / kadaiA / kadaiB / README.md / memo.md` が並んでいる。フォルダの信頼を尋ねるダイアログも別枠で示す

図22 正しく開けている状態。左のエクスプローラーの一番上に `HANDSON` と表示されます

✓ これができているOK

VSCode の左のエクスプローラーの一番上に `HANDSON` と表示され、その下に `_harness_kit`、`exercises`、`hints`、`kadaiA`、`kadaiB`、`README.md`、`memo.md` の7つが並んでいる（フォルダが先、ファイルが後の順で表示されます）。

`GENAI-NYUMON-HANDSON` と表示されている場合は、ひとつ上の階層を開いています。開き直してください。

Tips `CLAUDE.md` が見当たらないのは不備ではありません

AI 向けの設定ファイルの解説記事を読んだことがある方は、`CLAUDE.md` ・ `.claude` ・ `docs` が無いことに気付くかもしれません。これは意図的です。本研修は「設定が何も無い状態」から始め、途中で自分の手で設定を置いて、同じ AI の動きがどう変わるかを見比べます。`_harness_kit` の中にその材料が入っています。当日まで開かず置いておいてください。

動作確認

ここまでの設定がつながっているかを確認めます。確認は2つです。Claude Code が応答を返すことと、Java のアプリがビルドできることです。2つ目は時間がかかるため、必ず前日までに済ませてください。

01 Claude Code が応答するか

- 1 **handson** フォルダを開いた VSCode で、アクティビティバーの Claude Code アイコンを押します
右側または下側にパネルが開き、文字を入力する欄が現れます。

- 2 入力欄に次のとおり打って送信します

こんにちはと返してください

- 3 応答を待ちます

数秒で「こんにちは」を含む返事が返ってきます。初回はやや時間がかかることがあります。ファイルは何も変更されません。

スクリーンショット（撮影予定）

VSCode の Claude Code パネルで「こんにちはと返してください」と入力し、応答が返っている状態。
パネルの位置とアクティビティバーのアイコンが同じ画面に写っていること

図23 応答が返っている状態。ここまで確認できれば、Bedrock への接続を含めて設定は通っています

✓ これができているら OK

Claude Code のパネルに、日本語で「こんにちは」を含む返事が表示される。

応答が返らない、または認証に関するエラーが表示される場合は、SESSION 09 の設定と VSCode の再起動を確認してください。それでも解決しないときは SESSION 14 のトラブル 06 と 07 を見てください。

注意 ここで機密情報を入力しないでください

入力した文章と、AI が読んだファイルの中身は Amazon Bedrock へ送られます（図3）。動作確認では上の一文だけを送ってください。実在する社名、氏名、メールアドレス、電話番号、契約内容は入力しないでください。この決めごととは研修当日も同じです。

02 Java のアプリがビルドできるか

課題B のアプリは、動かす前に必要なライブラリをインターネットから取得します。この取得が初回だけ数分から十数分かかります。当日この時間を使うと演習が終わらないため、前日までに1回だけ済ませてください。2回目以降は手元に残ったものを使うため、数十秒で終わります。

1 VSCode のメニューから「表示」を開き、「ターミナル」を選びます

画面の下部にターミナルが開きます。開いている場所は `handson` フォルダです。

2 課題B のフォルダへ移動し、ビルドを実行します

ご自身の OS に合わせて、次のとおり入力してください。

Windows (PowerShell)

```
cd kadaiB
.\mvnw.cmd -q -DskipTests package
```

Mac (ターミナル)

```
cd kadaiB
./mvnw -q -DskipTests package
```

実行すると、ダウンロードの進捗が大量に流れます。止まっているように見えても待ってください。最後に `BUILD SUCCESS` と表示されれば成功です。 `-DskipTests` はテストを飛ばす指定で、ここでは依存ライブラリの取得だけが目的です。

スクリーンショット（撮影予定）

VSCode の下部ターミナルで mvnw のビルドが完了し、BUILD SUCCESS と経過時間が表示されている状態

図24 ビルドが成功した状態。 `BUILD SUCCESS` の行が出れば完了です

✓ これができればOK

ターミナルの出力の最後に `BUILD SUCCESS` と表示される。

`BUILD FAILURE` と出た場合は、その少し上に理由が書かれています。よくある原因は、Java のバージョンが 17 でないこと（SESSION 10）と、ライブラリの取得先へ通信できないこと（SESSION 02）の2つです。SESSION 14 のトラブル 13 も参照してください。

Tips `mvnw` とは何か

Java のプログラムを組み立てる道具を Maven と呼びます。 `mvnw` は Maven Wrapper の略で、配布プロジェクトに合ったバージョンの Maven を自動で用意して実行する起動用のスクリプトです。これが同梱されているため、Maven そのものを入れる必要がありません。 `./` や `.\` は「いま開いているフォルダの中にあるこれ」という意味です。

－ アプリを起動して画面まで見ておきたい場合（任意）

ビルドまでで十分ですが、当日の完成形を先に見ておきたい方は次を実行してください。

Windows は `.\mvnw.cmd spring-boot:run`、Mac は `./mvnw spring-boot:run` です。起動したらブラウザで `http://localhost:8080/equipments` を開くと、備品の一覧画面が表示されます。

止めるときは、ターミナルで `Ctrl + C` を押します。起動したままにしておくと、次に起動しようとしたときにポートが使用中で失敗します（SESSION 14 のトラブル 14）。

なお、このアプリには意図的な不具合が仕込まれています。表示がおかしい箇所があっても故障ではありません。当日それを見つけて直すのが課題B です。

SESSION 13

準備完了チェックリスト

研修前日までに、次の9項目すべてにチェックが入る状態にしてください。ひとつでも埋まらない項目がある場合は、当日の朝ではなく前日までに研修事務局へご連絡ください。当日の冒頭に環境確認の時間はありますが、そこで一から構築する時間はありません。

☐ SESSION 05・06

VSCode が起動する。「ようこそ」タブが表示される

☐ SESSION 07

メニューが「ファイル」「編集」のように日本語で表示される

☐ SESSION 08

拡張機能に「Claude Code for VS Code」（発行元 Anthropic）が入っている

☐ SESSION 08

左端のアクティビティバーに Claude Code のアイコンが出ている

☐ SESSION 09

環境変数4つを事務局配布の値で設定し、VSCode を再起動した

☐ SESSION 10

`java -version` の出力が `17.` から始まる

☐ SESSION 11

`genai-nyumon-handson.zip` を、日本語と空白を含まないパスに展開した

☐ SESSION 11

VSCode で `handson` フォルダを開くと、エクスプローラーの一番上に `HANDSON` と出る

☐ SESSION 12

Claude Code に「こんにちはと返してください」と送ると応答が返る

できていると当日が楽になる項目（必須ではありません）

`kadaiB` フォルダでビルドを1回実行し、`BUILD SUCCESS` を確認しておく、当日の起動待ちが数分から数十秒に縮みます（SESSION 12 の手順02）。20名が同時に取得を始めると回線が混み合うため、事前に済ませておくことを強くおすすめします。

トラブルシューティング

つまづきやすい14件を、症状から引ける形で並べています。上から読む必要はありません。ご自身の症状に近いものを探してください。どれにも当てはまらない場合は、画面のメッセージをそのままコピーして研修事務局へお送りください。

番号	症状	関連するセッション
01	VSCode がインストールできない	SESSION 05
02	Mac で「開発元を検証できないため開けません」と出る	SESSION 06
03	拡張機能の検索に Claude Code が出てこない	SESSION 08
04	日本語化したのに表示が英語のまま	SESSION 07
05	アクティビティバーに Claude Code のアイコンが出ない	SESSION 08
06	Claude Code が認証エラーになる、応答が返らない	SESSION 09・12
07	環境変数を設定したのに反映されない	SESSION 09
08	社内プロキシ環境で通信がブロックされる	SESSION 02
09	<code>java -version</code> が「コマンドが見つかりません」になる	SESSION 10
10	<code>java -version</code> が 17 以外を表示する	SESSION 10
11	展開したのに handson フォルダが見つからない	SESSION 11
12	Mac で <code>./mvnw</code> が Permission denied になる	SESSION 12
13	ビルドが依存ライブラリの取得で失敗する	SESSION 12
14	ポート 8080 が使用中でアプリが起動しない	SESSION 12

01 VSCode がインストールできない

考えられる原因

管理者権限が必要な System Installer を実行している、または会社の資産管理ソフトがインストールを止めています。

対処

1. ダウンロードページの Windows の欄で「Select a download...」を開き、**User Installer** (x64) を選び直します。ユーザーのフォルダ配下に入るため、管理者権限を求められません
2. それでも止まる場合、インストール自体が組織のポリシーで制限されています。情報システム部門へ VSCode の導入可否をご確認ください
3. Mac で「アプリケーション」フォルダへ移動できない場合は、ホームフォルダの直下に置いても動作します

02 Mac で「開発元を検証できないため開けません」と出る

考えられる原因

macOS が、インターネットから取得したアプリの実行を一度止める仕組みです。故障ではありません。

対処

1. 警告のダイアログを「OK」で閉じます
2. アップルマークから「システム設定」を開き、左の一覧から「プライバシーとセキュリティ」を選びます
3. 画面を下へたどると「"Visual Studio Code" は開発元を確認できないため、使用がブロックされました」という表示があります
4. その右の「このまま開く」を押し、確認のダイアログでもう一度「開く」を押します。
管理者パスワードを求められたら入力します

公式サイト（ code.visualstudio.com ）から入手したことを確認したうえで実行してください。

03 拡張機能の検索に Claude Code が出てこない

考えられる原因

VSCode が古い、または拡張の配布元へ通信できていません。Claude Code 拡張は VSCode 1.94.0 以上を必要とします。

対処

1. メニューの「ヘルプ」から「更新の確認」を選び、最新版へ更新してから検索し直します
2. 更新後もヒットしない場合、`marketplace.visualstudio.com` への通信が制限されている可能性があります。SESSION 02 の一覧を情報システム部門へ共有し、疎通の確認を依頼してください
3. 検索欄に `anthropic.claude-code` と識別子を直接入力すると、名前の揺れの影響を受けずに探せます

04 日本語化したのに表示が英語のまま

考えられる原因

再起動していない場合がほとんどです。Mac では、ウインドウを閉じてでもアプリが終了していません。

対処

1. Windows は VSCode を閉じてから開き直します。Mac は `Cmd` + `Q` で完全に終了してから開き直します
2. それでも変わらない場合、コマンドパレット (`Ctrl` / `Cmd` + `Shift` + `P`) で `Configure Display Language` を実行し、「日本語 (ja)」を選び直します
3. 一覧に「日本語」が出てこない場合は、言語パックが入っていません。SESSION 07 の方法Bで拡張機能から入れてください

05 アクティビティバーに Claude Code のアイコンが出ない

考えられる原因

拡張が有効になっていない、またはアイコンが「…」の中に隠れています。

対処

1. アクティビティバーの下部にある「…」を押し、一覧に Claude Code がないか確認します
2. VSCode を再起動します。インストール直後は反映が遅れることがあります
3. 拡張機能の一覧で「Claude Code for VS Code」を開き、「有効にする」ボタンが出ていないか確認します。出ていれば押します
4. コマンドパレットで `Claude` と入力し、関連するコマンドが出れば拡張自体は入っています。その場合はアイコンの表示だけの問題です

考えられる原因

環境変数の値の取りこぼし、VSCode の再起動忘れ、Bedrock への通信の遮断のいずれかです。上から順に確認してください。

対処

1. 環境変数を設定したあとに VSCode を **完全に終了して** 起動し直したか確認します。これが最も多い原因です
2. 新しく開いたターミナルで `echo` を実行し、`CLAUDE_CODE_USE_BEDROCK` が `1` を返すか確認します (SESSION 09 の確認方法)
3. 認証情報を、案内の文面から前後の空白ごとコピーしていないか確認します。分からない場合は一度手で打ち直してください
4. 引用符が全角 (`”`) になっていないか確認します
5. ここまで問題がなければ、`bedrock-runtime` への通信が遮断されている可能性があります。SESSION 02 の一覧を情報システム部門へ共有してください

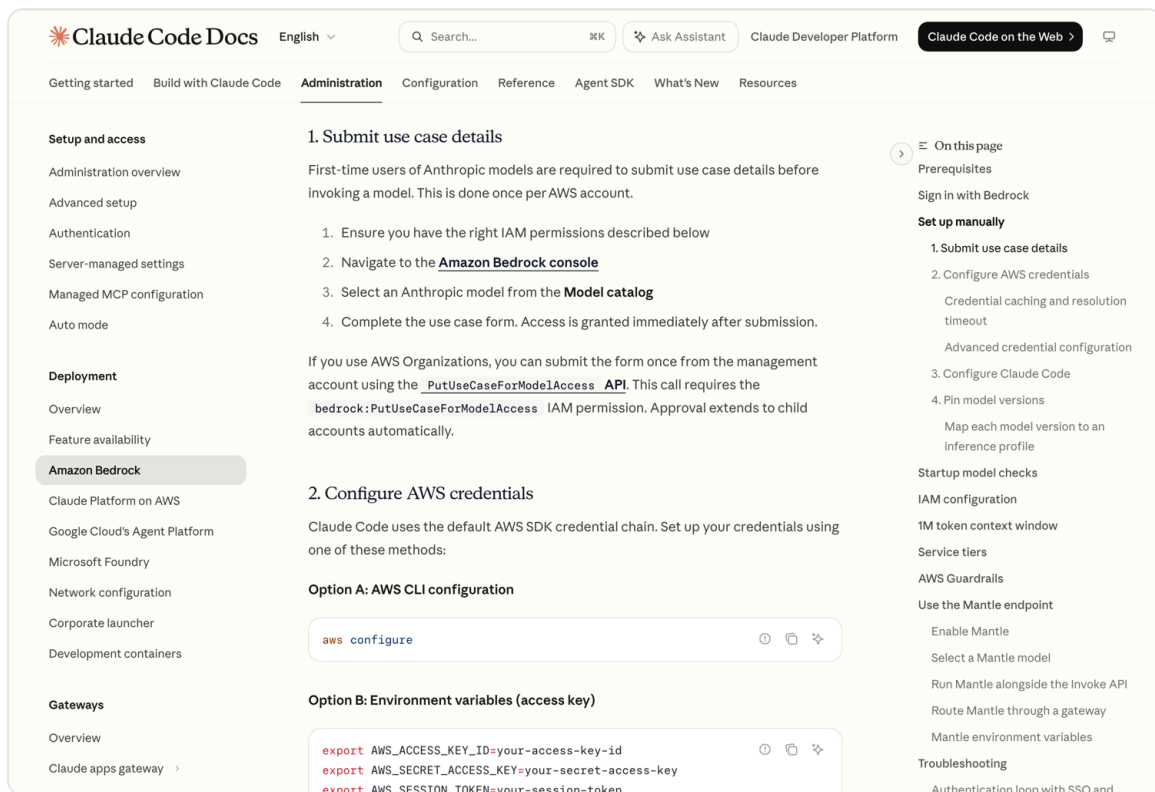


図25 公式ドキュメントの認証情報の設定方法。本研修では、環境変数で渡す方法 (Option B) を使います

07 環境変数を設定したのに反映されない

考えられる原因

環境変数は、設定したあとに**新しく起動したアプリ**にだけ届きます。すでに開いていたアプリには反映されません。

対処

1. ターミナル（PowerShell）を閉じて、新しく開き直してから `echo` で確認します
2. VSCode も完全に終了してから開き直します。Windows はタスクバーに残っていないか、Mac は `Cmd + Q` で終了したかを確認してください
3. Windows で `setx` の実行時に「成功: 指定した値は保存されました。」が出ていたか確認します。出ていなければ登録されていません
4. Mac で追記先を間違えている可能性があります。ターミナルで `tail -5 ~/.zshrc` を実行し、末尾に `export CLAUDE_CODE_USE_BEDROCK=1` の行があるか確認してください

スクリーンショット（撮影予定）

環境変数が反映されていないときのターミナル出力（echo が空行を返している状態）と、
反映されているときの出力を上下に並べて対比する

図T1 反映されていないときと、されているときの出力の違い

08 社内プロキシ環境で通信がブロックされる

考えられる原因

社内から外部へ出る通信が、プロキシサーバーを経由する構成になっています。VSCode 側にもその設定が要ります。

対処

1. 情報システム部門にプロキシのアドレスを確認します
2. VSCode の設定（`Ctrl / Cmd + ,`）を開き、検索欄に `proxy` と入力して「Http: Proxy」に確認したアドレスを入力します
3. ターミナルからのビルドでも必要な場合は、環境変数 `HTTPS_PROXY` と `HTTP_PROXY` に同じアドレスを設定します。設定方法は SESSION 09 と同じ形です
4. プロキシで通信の中身を検査している環境（SSL インспекション）では、証明書の追加が必要になることがあります。この場合は受講者側では解決できないため、情報システム部門へご相談ください

09

java -version が「コマンドが見つかりません」になる**考えられる原因**

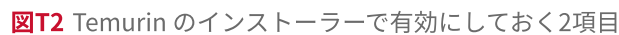
JDKが入っていないか、入っていても場所がOSに登録されていません。Windowsのインストーラーで `Add to PATH` を有効にしなかった場合によく起きます。

対処

1. ターミナル (PowerShell) を閉じて開き直してから、もう一度実行します。インストール直後は開いていた画面に反映されません
2. Windows は Temurin のインストーラーをもう一度実行し、「変更」を選んで `Set JAVA_HOME variable` と `Add to PATH` を有効にします
3. Mac で PKG からインストールしたのに見つからない場合は、ターミナルで `/usr/libexec/java_home -v` を実行し、17 が一覧にあるか確認します

スクリーンショット (撮影予定)

Windows のインストーラーのカスタムセットアップ画面で、Set JAVA_HOME variable と Add to PATH を有効にした状態

 **T2** Temurin のインストーラーで有効にしておく2項目

10

java -version が 17 以外を表示する**考えられる原因**

複数の JDK が入っていて、17 以外のほうが先にみつかっています。

対処 (Windows)

1. スタートメニューで「環境変数」と検索し、「システム環境変数の編集」を開きます
2. 「環境変数」ボタンを押し、`JAVA_HOME` の値を JDK 17 のインストール先に設定します
3. `Path` を編集し、`%JAVA_HOME%\bin` を一覧の上のほうへ移動します
4. PowerShell を開き直して `java -version` を再実行します

対処 (Mac)

1. ターミナルで `/usr/libexec/java_home -V` を実行し、17 が一覧にあることを確認します
2. 次を実行して設定ファイルに追記します

```
echo 'export JAVA_HOME=$(/usr/libexec/java_home -v 17)' >> ~/.zshrc
```

ターミナルを開き直してから `java -version` を再実行してください。

11 展開したのに handson フォルダが見つからない

考えられる原因

展開先を確認せずに進めた、または同じ名前のフォルダが二重になっています。

対処

1. ダウンロードフォルダで `genai-nyumon-handson.zip` を探し、右クリックから「すべて展開」（Mac はダブルクリック）でやり直します
2. できたフォルダを開き、中に `handson` があることを確認します。`genai-nyumon-handson` の中の `genai-nyumon-handson` の中の `handson` という三重構造になっている場合は、いちばん内側の `handson` を VSCode で開きます
3. VSCode で開いたあと、左のエクスプローラーの一番上が `HANDSON` になっていれば正しい階層です

12 Mac で `./mvnw` が Permission denied になる

考えられる原因

ZIP から展開したときに、ファイルを実行してよいという印が外れることがあります。

対処

ターミナルで `kadaiB` フォルダに移動し、次を実行してから、もう一度ビルドを試してください。

```
chmod +x mvnw
```

それでも動かない場合は `sh mvnw -q -DskipTests package` のように `sh` を前に付けて実行できます。

13 ビルドが依存ライブラリの取得で失敗する

考えられる原因

ライブラリの配布元へ通信できていないか、途中で中断されて壊れたファイルが残っています。

対処

1. 出力の中に `Could not transfer artifact` や `Connection timed out` があれば通信の問題です。
`repo.maven.apache.org` への通信の許可を情報システム部門へご依頼ください (SESSION 02)
2. プロキシ環境の場合は、トラブル 08 の手順で環境変数を設定してから再実行します
3. 一度中断してから再実行して失敗する場合、取得しかけのファイルが残っています。
ホームフォルダの `.m2` フォルダを削除してから再実行すると直ることがあります
4. 回線が遅い環境では、単に時間がかかっているだけのこともあります。10分程度は待ってみてください

スクリーンショット (撮影予定)

依存ライブラリの取得に失敗したときのターミナル出力。BUILD FAILURE と `Could not transfer artifact` の行を枠で囲んで示す

図T3 取得に失敗したときの出力。どの行を見れば原因が分かるかを示します

14 ポート 8080 が使用中でアプリが起動しない

考えられる原因

前に起動したアプリが終了していないか、別のソフトが同じ番号を使っています。出力に `Web server failed to start. Port 8080 was already in use.` と表示されます。

対処

1. 前に起動したターミナルが残っていないか確認し、そこで `Ctrl + C` を押して止めます
2. VSCode の下部にターミナルが複数開いている場合、右側の一覧から前のものを選んで止めてください
3. 見当たらない場合は、Windows は `netstat -ano | findstr :8080`、Mac は `lsof -i :8080` で使用中の相手を確認できます
4. どうしても解放できない場合は、PC を再起動するのが早いです

スクリーンショット (撮影予定)

ポートが使用中で起動に失敗したときのターミナル出力。Port 8080 was already in use の行を枠で囲んで示す

図T4 ポートが埋まっているときの出力

参考リンクと当日の持ち物

本ガイドで参照した公式ドキュメントと、当日お持ちいただくものです。準備が終わっていれば、当日はPCを持ってお越しいただくだけです。

01 当日の持ち物

持ち物	備考
ノート PC	本ガイドのセットアップを済ませたもの。電源アダプタも合わせてお持ちください
接続情報の控え	設定済みであれば当日使いませんが、再設定が必要になったときのために手元にあると安心です
筆記用具	演習中に気付いたことを書き留めるために使います。手元のファイルに書いても構いません

当日の進め方について

研修は4時間で、うち半分以上が手を動かす時間です（図2）。分からないところで止まっても、講師が回りますので、そのままにせずお声がけください。演習の手順書は配布ファイルの `exercises` フォルダに入っています。当日はそれを見ながら進めます。

02 本ガイドで参照した公式ドキュメント

内容	リンク
Visual Studio Code 公式サイト	https://code.visualstudio.com/
VSCode のダウンロード	https://code.visualstudio.com/Download
VSCode の表示言語の切り替え	https://code.visualstudio.com/docs/configure/locales
日本語化パックの配布ページ	Japanese Language Pack for Visual Studio Code
Claude Code 拡張の配布ページ	Claude Code for VS Code
Claude Code を VS Code で使う	https://docs.claude.com/en/docs/claude-code/vs-code
Claude Code の Amazon Bedrock 接続	https://docs.claude.com/en/docs/claude-code/amazon-bedrock
Claude Code の設定と環境変数	https://docs.claude.com/en/docs/claude-code/settings
Amazon Bedrock 製品ページ	https://aws.amazon.com/jp/bedrock/
Eclipse Temurin JDK 17	https://adoptium.net/temurin/releases/?version=17
Temurin のインストール手順	https://adoptium.net/installation/
Maven Wrapper	https://maven.apache.org/wrapper/
Spring Boot リファレンス	https://docs.spring.io/spring-boot/index.html

03 用語のおさらい

本ガイドに出てきた言葉のうち、当日も繰り返し使うものをまとめます。

用語	意味
アクティビティバー	VSCode の画面いちばん左にある、縦に並んだアイコンの列。Claude Code はここから開きます
エクスプローラー	アクティビティバーの一番上のアイコンで開く、フォルダの中身を一覧する領域
コマンドパレット	<code>Ctrl / Cmd + Shift + P</code> で開く入力欄。VSCode のあらゆる操作を名前で呼び出せます
ターミナル	コマンドを打って PC を操作する画面。VSCode の中にも内蔵されています
環境変数	OS に登録しておく名前付きの設定値。アプリが起動時に読み取ります
拡張機能	VSCode に機能を足す部品。Claude Code も日本語化パックもこれです
JDK	Java のプログラムを組み立てて動かすための土台一式
ビルド	人が書いたコードを、機械が実行できる形に組み立てること
Amazon Bedrock	AWS 上で Claude などのモデルを呼び出すための入口。本研修の接続先です
Claude Sonnet 4.6	本研修で使うモデルの名前。Amazon Bedrock 経由で呼び出します

準備の途中でご不明な点がございましたら、案内メールに記載の問い合わせ先までご連絡ください。

Givery, Inc.

