

協栄産業株式会社 様 | 生成AI実務研修(2日間)

事前セットアップ手順 (Windows / Mac 共通)

研修日の3営業日前までに、本書の手順で環境を整えてください

アジェンダ

インストールするもの(上から順に)

1. VS Code (エディタ)
2. Node.js (Claude Code の導入に使用)
3. Python 3.12 以上
4. Claude Code CLI (npm で導入) + ログイン
5. uv / uvx (Spec Kit の実行に使用)

準備

6. 配布 ZIP (kyos-handson.zip) の展開と VS Code で開く
7. 動作確認チェック

はじめに(前提と完了の判定)

対象・環境

受講者 (Claude Code の既使用・初使用どちらも)。OS は Windows 11 / macOS 13 以降。所要 30~60 分。

Claude Code は **CLI 版**を使います。VS Code の統合ターミナル (メニュー「ターミナル」→「新しいターミナル」) から `claude` で起動します。拡張機能版はメインにしません。

完了の判定

最後の「動作確認チェック」がすべて OK になれば準備完了です。つまりいたら「トラブルシューティング」を見て、解決しなければ研修事務局へご連絡ください。

配布 ZIP (`kyos-handson.zip`) は事前にお送りします。GitHub アカウントは必須ではありません。

STEP 1

VS Code のインストール

VS Code のインストール

① インストーラーのダウンロード

公式サイト code.visualstudio.com/Download を開き、お使いの OS のボタンを押します。

Visual Studio Code Features Docs Release Notes Blog Learn Events Resources Download

Rewatch VS Code Live at MS Build 2026

Download Visual Studio Code

Free and built on open source. AI agents, Git, debugging, and extensions included.

Windowsはこちら

↓ Windows
Windows 10, 11

Linuxはこちら

↓ .deb
Debian, Ubuntu

↓ .rpm
Red Hat, Fedora, SUSE

Macはこちら

↓ Mac
macOS 12.0+

User Installer [x64](#) [Arm64](#)
System Installer [x64](#) [Arm64](#)
.zip [x64](#) [Arm64](#)
CLI [x64](#) [Arm64](#)

.deb [x64](#) [Arm32](#) [Arm64](#)
.rpm [x64](#) [Arm32](#) [Arm64](#)
.tar.gz [x64](#) [Arm32](#) [Arm64](#)
Snap [Snap Store](#)
CLI [x64](#) [Arm32](#) [Arm64](#)

.dmg [Intel chip](#) [Apple silicon](#) [Universal](#)
CLI [Intel chip](#) [Apple silicon](#)

VS Code のインストール

② インストールと確認

ダウンロードしたインストーラーを開き、画面の指示に従って進めます。

Windows:「PATH への追加」にチェック。macOS:アプリを Applications に移動し、コマンドパレット (Cmd+Shift+P) で「Shell Command: Install 'code' command in PATH」を実行。

統合ターミナルで次を実行し、バージョンが表示できれば OK です。

```
code --version
```

STEP 2

Node.js のインストール

Node.js のインストール

① LTS 版をダウンロードして導入

公式サイト nodejs.org/ja から LTS 版を入手し、画面の指示どおりに (既定設定のまま) インストールします。



The screenshot shows the Node.js website in Japanese. The main heading is "どこでもJavaScript を使おう" (Use JavaScript anywhere). Below it, there is a description of Node.js as a cross-platform JavaScript runtime. A red box highlights the "LTS 版を入手" (Get LTS version) button, which is a green button with the text "Node.js@を入手". Below this, there is a "セキュリティサポート" (Security support) section for EOL versions. On the right side, there is a code editor showing a simple HTTP server example in JavaScript. At the bottom, there are logos of partner companies and a note that Node.js is supported by many of them.

```
1 // server.mjs
2 import { createServer } from 'node:http';
3
4 const server = createServer((req, res) => {
5   res.writeHead(200, { 'Content-Type': 'text/plain' });
6   res.end('Hello World!\n');
7 });
8
9 // starts a simple http server locally on port 3000
10 server.listen(3000, '127.0.0.1', () => {
11   console.log('Listening on 127.0.0.1:3000');
12 });
13
14 // run with `node server.mjs`
```

確認: `node --version` と `npm --version` でバージョンが出れば OK。

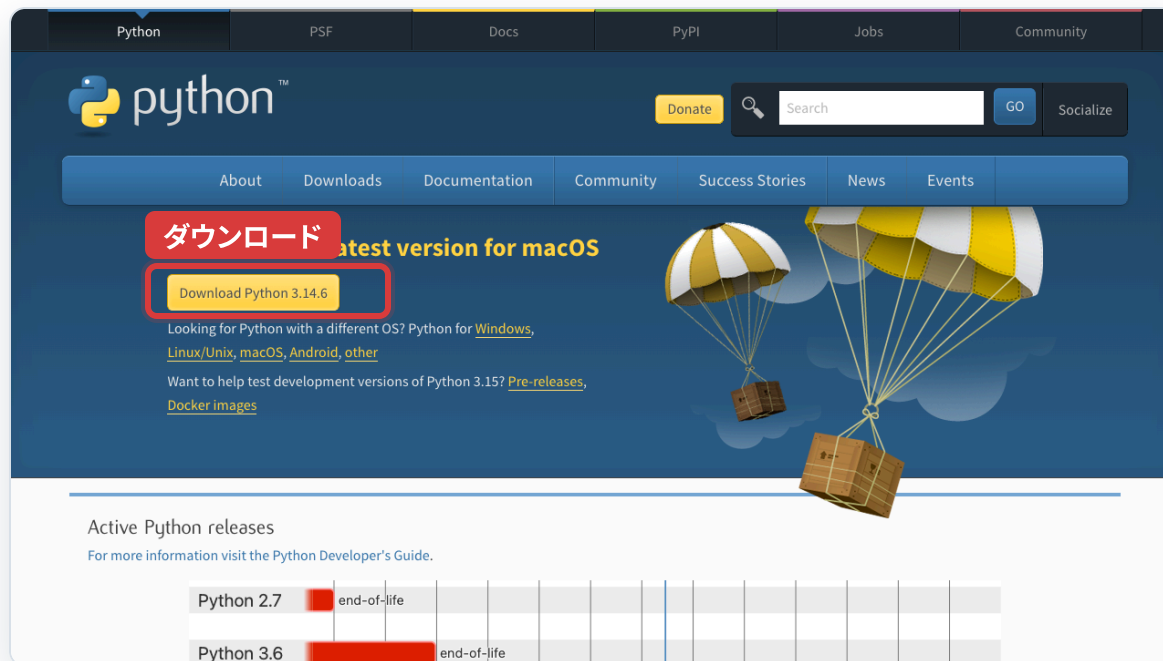
STEP 3

Python のインストール

Python のインストール(3.12 以上)

① ダウンロード

公式サイト python.org/downloads から黄色のダウンロードボタンで取得します。



Windows は要注意: インストーラー最初の画面で「Add python.exe to PATH」に必ずチェックを。macOS は `brew install python@3.12` でも可。

Python のインストール(3.12 以上)

② 確認

統合ターミナルで次を実行し、3.12 以上が表示されれば OK です。

```
python --version      # macOS は python3 --version  
python -m pytest --version  # pytest が無ければ: pip install pytest
```

研修では仕様駆動 × テスト駆動で `pytest` を使います。上の2つが通れば Python 側の準備は完了です。

STEP 4

Claude Code CLI の導入とログイン

Claude Code CLI のインストール

① npm で導入

VS Code の統合ターミナル (または OS 標準のターミナル) で次を実行します。

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

確認: `claude --version` でバージョン (例 `2.x.x`) が出れば OK。最新手順は code.claude.com/docs/ja/setup。

Claude Code のログイン

② アカウントで認証

API キーは使いません。ターミナルで `claude` を起動すると初回にブラウザが開くので、**Pro / Max** いずれかが有効な **Claude アカウント** でログインします。

```
claude # 初回はブラウザでログイン画面が開く
```

ログイン後、ターミナルに対話画面（入力欄）が出れば OK。研修では Sonnet 4.6 を標準に、用途で Opus 4.7 / Haiku 4.5 を切り替えます。

STEP 5

uv / uvx のインストール

uv / uvx のインストール

① 導入 (Spec Kit の実行に使用)

Windows (PowerShell)

```
powershell -ExecutionPolicy ByPass -c "irm  
https://astral.sh/uv/install.ps1 | iex"
```

macOS / Linux

```
curl -Lsf https://astral.sh/uv/install.sh | sh
```

確認

ターミナルを開き直してから `uvx --version` でバージョンが出れば OK。

スクリプト直実行が社内ポリシーで難しい場合は `pip install uv` でも代替できます。公式 docs.astral.sh/uv

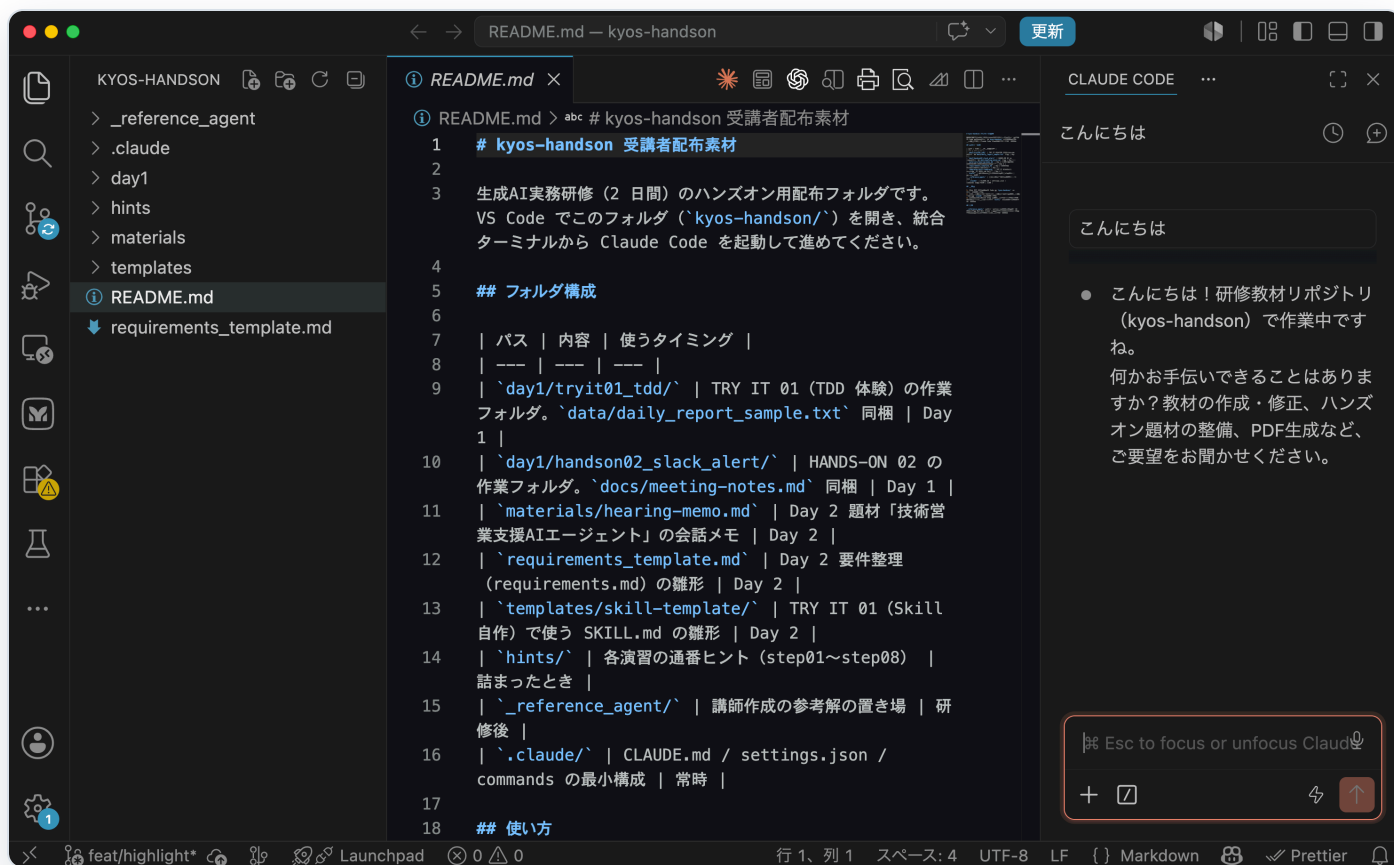
STEP 6

配布フォルダの展開

配布 ZIP を展開して VS Code で開く

① 解凍 → 「フォルダーを開く」

事前配布の `kyos-handson.zip` を作業フォルダ (例 `C:\work\kyos`) に解凍し、VS Code のメニュー「ファイル」→「フォルダーを開く」で `kyos-handson` を開きます。右に Claude Code を表示できます。



動作確認チェック(研修当日の朝までに)

確認コマンド / 操作	OK の状態
<code>code --version</code>	バージョンが表示される
<code>node --version</code> / <code>npm --version</code>	LTS 版 (v22 系以上) が表示される
<code>python --version</code>	3.12 以上が表示される
<code>claude --version</code>	バージョンが表示される (ログイン済み)
<code>uvx --version</code>	バージョンが表示される
配布 ZIP を VS Code で開ける	<code>README.md</code> が見える

仕上げ: `kyos-handson` で `claude` を起動し「`hello.py` を作って実行して」と打ち、`Hello` が出れば「作って・動かす」まで確認完了です。

ハンズオン題材(ブラウザで動くアプリ)の開き方

研修の題材アプリ(部品在庫ダッシュボード・提案アシスタント)は `index.html` を開くと動きます。表示が崩れる場合は、そのフォルダで簡易サーバを起動してください。

```
cd kyo5-handson/apps/parts-dashboard  
python -m http.server 8000      # ブラウザで http://localhost:8000/ を開く
```

テストの実行: ブラウザ題材は `node --test tests/`、Spec Kit / Kiro / TDD は `python -m pytest tests/ -v` です(当日に解説します)。

トラブルシューティング

コマンドが見つからない(PATH)

VS Code とターミナルを全部閉じて開き直す。Windows は `where claude`、macOS は `which claude` で所在を確認。

権限エラーで失敗

Windows は PowerShell を「管理者として実行」。macOS は `sudo` を避け npm 公式手順でユーザー領域に変更。

社内プロキシで失敗

環境変数 `HTTPS_PROXY` / `HTTP_PROXY` を設定。npm は `npm config set proxy ...`。SSL 検査環境は社内ルート証明書の取り込みが必要な場合あり。

会社PCで管理者権限がない

情報システム部門にインストール申請。間に合わない場合は研修事務局へ事前連絡。

解決しないときは

研修事務局 (Givery 加藤) までご連絡ください。研修当日の朝までに解決できない場合は、当日冒頭のオープニング中に個別対応します。

研修開始時に手元にあるもの: 本書 (事前セットアップ) / オリエンテーション座学編 / ハンズオンガイド Day1・Day2 / kyos-handson.zip 。