



Givery, Inc.

PRE-SEMINAR SETUP · 2026 年 5 月版

事前セットアップガイド

パーソルクロステクノロジー株式会社様 CA企画 / エージェントティックエンジニアリング

提供	Givery 株式会社
配布先	パーソルクロステクノロジー株式会社 御中
対象	2026 年 5 月 21 日(木) 19:00-21:00 セミナー参加者
所要	20~40 分(環境による)
版数	v1.0(2026-05-09)

セミナー当日のハンズオンを 45 分でやり切るため、ZOOM 接続の前に手元で済ませておく作業をまとめています。前日までに完了しておいてください。

1. このガイドの位置付けと所要時間

本書は当日 19:55 から始まるハンズオン枠でつまづかないための準備手順です。VS Code と Claude Code を入れ、配布 ZIP を解凍して、サンプルログを 1 度読み込ませるところまで前日に終わらせます。20~40 分を見ておけば足りません。

所要時間	20~40 分
必要なライセンス	Claude Code(Pro 以上推奨)または GitHub Copilot
OS	macOS / Windows 10・11 / Ubuntu 22.04 以降
対象者	コードを書くエンジニア、LLM API 利用経験者

1.1 当日の流れと本書の対応

時刻	内容	本書の参照
19:00-19:40	オープニングと概念整理	該当なし
19:40-19:55	講師による Before デモ	該当なし
19:55-20:40	ハンズオン	本書 §3~§6 の準備が必須
20:40-21:00	振り返り・Q&A	該当なし

前日までにここまで終えてください

VS Code 起動、Claude Code 起動して ` /help ` が出る、配布 ZIP を解凍して ` claude ` コマンドが起動できる、の 3 点。どれか落ちると当日の 45 分でほぼ追いつきません。

2. 環境前提

手元のマシンが下の表を満たしているかを最初に確認してください。会社支給機でプロキシや EDR が入っている場合は §7 のチェックも先に通します。

項目	必須	確認コマンド
OS	macOS 13 以降 / Windows 10・11 / Ubuntu 22.04 以降	—
Node.js	v20 LTS 以上 (Claude Code が必要)	<code>node -v</code>
Git	v2.30 以上	<code>git --version</code>
jq	v1.6 以上 (ハンズオン §4 で使用)	<code>jq --version</code>
ターミナル	zsh / bash / PowerShell 7 / WSL2 のいずれか	—
ZOOM	最新版クライアント	—

2.1 Node.js が入っていない場合

macOS は Homebrew、Windows は [fnm](#)、Ubuntu は [nvm](#) 経由が手堅い。バージョン固定が要らないならどれでも構いません。

```
# macOS
brew install node@20
echo 'export PATH="/opt/homebrew/opt/node@20/bin:$PATH"' >> ~/.zshrc

# Windows (PowerShell 7 + winget)
winget install Schniz.fnm
fnm install 20
fnm default 20

# Ubuntu / WSL2
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
nvm install 20
```

2.2 jq が入っていない場合

```
# macOS
brew install jq

# Windows
winget install jqlang.jq

# Ubuntu / WSL2
sudo apt-get install -y jq
```

3. VS Code のインストール

セミナーは VS Code を前提に進めます。すでに入っているならスキップして §4 へ。

1. <https://code.visualstudio.com/Download> から OS に合うインストーラーを取得
2. インストーラーの指示に従って完了
3. 起動して、コマンドパレット (macOS は `Cmd+Shift+P`、Windows / Linux は `Ctrl+Shift+P`) から `Shell Command: Install 'code' command in PATH` を実行 (macOS のみ)
4. ターミナルから `code --version` で出力が出れば完了

3.1 推奨拡張

拡張	役割	必須
Claude Code	VS Code と Claude Code の連携、差分表示	強く推奨
GitHub Copilot	Copilot 利用時の代替経路	任意
Remote - WSL	Windows で WSL2 を使う場合	Windows のみ

4. Claude Code のインストールと初期化

公式 CLI を Node.js 経由で入れ、サインインまで終わらせます。サインインのブラウザ認証だけは Web 接続が要るので、社内ネットワークで詰まりがちな環境では §7 を先に確認してください。

4.1 インストール

```
# 公式 npm パッケージ。グローバルに入れる
npm install -g @anthropic-ai/claude-code

# バージョン確認 (2.1 系が出ていれば OK)
claude --version
```

2026 年 5 月時点の最新

Claude Code 2.1.133(2026-05-07 時点の最新)。Skills(SKILL.md)が SKILL.md ベースで提供され、Hooks の async 実行も v2.1.121(2026-04-28)以降で対応。出典:[公式 Changelog](#)。

4.2 初回サインイン

```
# 任意の作業ディレクトリで起動
mkdir -p ~/Desktop/agentic-seminar
cd ~/Desktop/agentic-seminar
claude

# 起動後にブラウザが開き、Anthropic のサインイン画面に飛ぶ
# Pro 以上のサブスク、または API キー利用で進む
```

サインイン完了すると、ターミナルに対話プロンプトが出ます。試しに `/help` を打って、コマンド一覧が表示されれば成功です。

4.3 GitHub Copilot で代替する場合

Claude Code が手配できない方は GitHub Copilot を VS Code 拡張で動かす形でも参加できます。ただし当日のハンズオンの主目標は「Hooks / Skills / サブエージェントの体験」であり、これらは Claude Code 固有機能のため、Copilot 環境では一部のステップを「読む」だけになります。可能であれば Claude Code 側を整えてください。

業務メールでサインインしないでください

本セミナーは派遣スタッフ向けで、申込時に「個人のメールアドレス」が指定されています。Claude Code のサインインも同じく個人アカウントを使ってください。

5. 配布 ZIP の解凍と中身確認

セミナー前日に配布される受講者向け ZIP を解凍し、ディレクトリ構造を確認します。当日この構造のまま開きます。

5.1 ZIP の中身

```
log-review-agent/  
├── .claude/  
│   ├── settings.json      # hooks 雛形 (中身は当日埋める)  
│   ├── agents/log-reader.md # サブエージェント定義  
│   └── skills/summarize-log/ # Skill 定義  
├── hooks/  
├── sample-logs/          # 5xx 混じりの Nginx 風アクセスログ等  
├── hints/  
├── solutions/            # 完成版 (リカバリ用)  
├── CLAUDE.md  
└── README.md
```

5.2 解凍と起動チェック

1. 配布 ZIP (`log-review-agent_受講者用.zip`) をダウンロード
2. ホームの好きな場所に解凍 (パスに日本語・空白を含めない方が安全)
3. VS Code で開く: `code log-review-agent`
4. 同じディレクトリで Claude Code を起動: `cclaude`
5. プロンプトに `/agents` と打って、`log-reader` が一覧に出ること
6. 続けて `/skills` と打って、`summarize-log` が一覧に出ること

確認に時間をかけすぎない

動作確認はここまで。ハンズオン本番のステップは当日進めます。中身を先に書くと「考える」体験が消えるので開けないでください。`solutions/` も同様、当日詰まったときだけ開く想定です。

6. 起動チェック(前日 5 分)

下の 6 項目をチェックしてください。1 つでも × があれば §8 のトラブル対応へ。

01

node -v で v20 系が表示される

02

git --version で 2.30 以上が表示される

03

jq --version で 1.6 以上が表示される

04

code --version で出力が出る

05

claude --version で 2.1 系が表示される

06

配布 ZIP を解凍したディレクトリで **claude** を起動し、**/agents** と **/skills** がそれぞれ 1 件以上応答する

7. 社内環境制約への対応

プロキシ、SSL 証明書、EDR、社内ファイアウォールに引っかかる典型例を先に潰しておきます。会社支給端末でセミナーに参加する方はここを必ず読んでください。

症状	原因の見当	対応
<code>npm install</code> が SSL エラーで止まる	社内中継のオレオレ証明書	会社配布の root 証明書を <code>NODE_EXTRA_CA_CERTS</code> に通す
<code>claude</code> 起動でブラウザ認証が完了しない	localhost コールバックがブロック	個人端末で実行するか、IT 部門に開放を依頼
WebFetch / 公式ドキュメント取得が失敗	外向きプロキシ未設定	<code>HTTPS_PROXY</code> 環境変数を設定
<code>code .</code> が PATH に通らない	VS Code の PATH 連携未実行	コマンドパレットから <code>Install 'code' command</code>
EDR が <code>.sh</code> 実行をブロック	セキュリティポリシー	個人端末で実行を推奨

就業先のメールアドレスは使わない

申込ページにも明記されています。個人メール、個人端末で参加するのが最も詰まりが少ないルートです。

8. 当日詰まったらどうするか

500 名規模のオンラインセミナーで、講師 1 人で全員を救うのは難しいです。詰まったら次の順に進めてください。

1

まず動かす。 `solutions/` 配下の完成版を `.claude/` に上書きコピーして先のステップに進む。考えるのは後でできます。

2

ヒントを開く。 `hints/stepNN_xxx.md` に参考プロンプトと比較観点があります。各ステップ末尾の参照行に番号を書いています。

3

ZOOM のチャットに質問を投げる。CA企画事務局がモデレーション、講師は Q&A 枠(20:40～)でまとめて回答します。

4

後追い学習。終了後に `solutions/` と `hints/README.md` を見ながら自分で再現すれば 30 分で同じ動作にたどり着きます。

解凍前に `solutions/` を開かない

答えを先に見るとハンズオンの「考える→書く」が機能しません。当日のリカバリ用と割り切ってください。

参考リンク

- [セミナー申込ページ\(パーソルクロステクノロジー CA企画\)](#)
- [Claude Code クイックスタート](#)
- [Claude Code Changelog](#)
- [VS Code ダウンロード](#)
- [fnm\(Node.js バージョンマネージャ\)](#)
- [nvm](#)
- [jq](#)
- [GitHub Copilot 公式ドキュメント](#)



エージェンティックエンジニアリング(パーソルクロステクノロジー様 CA企画)

2026 年 5 月 21 日(木) 19:00-21:00 / 提供:Givery 株式会社

© Givery, Inc. All Rights Reserved.