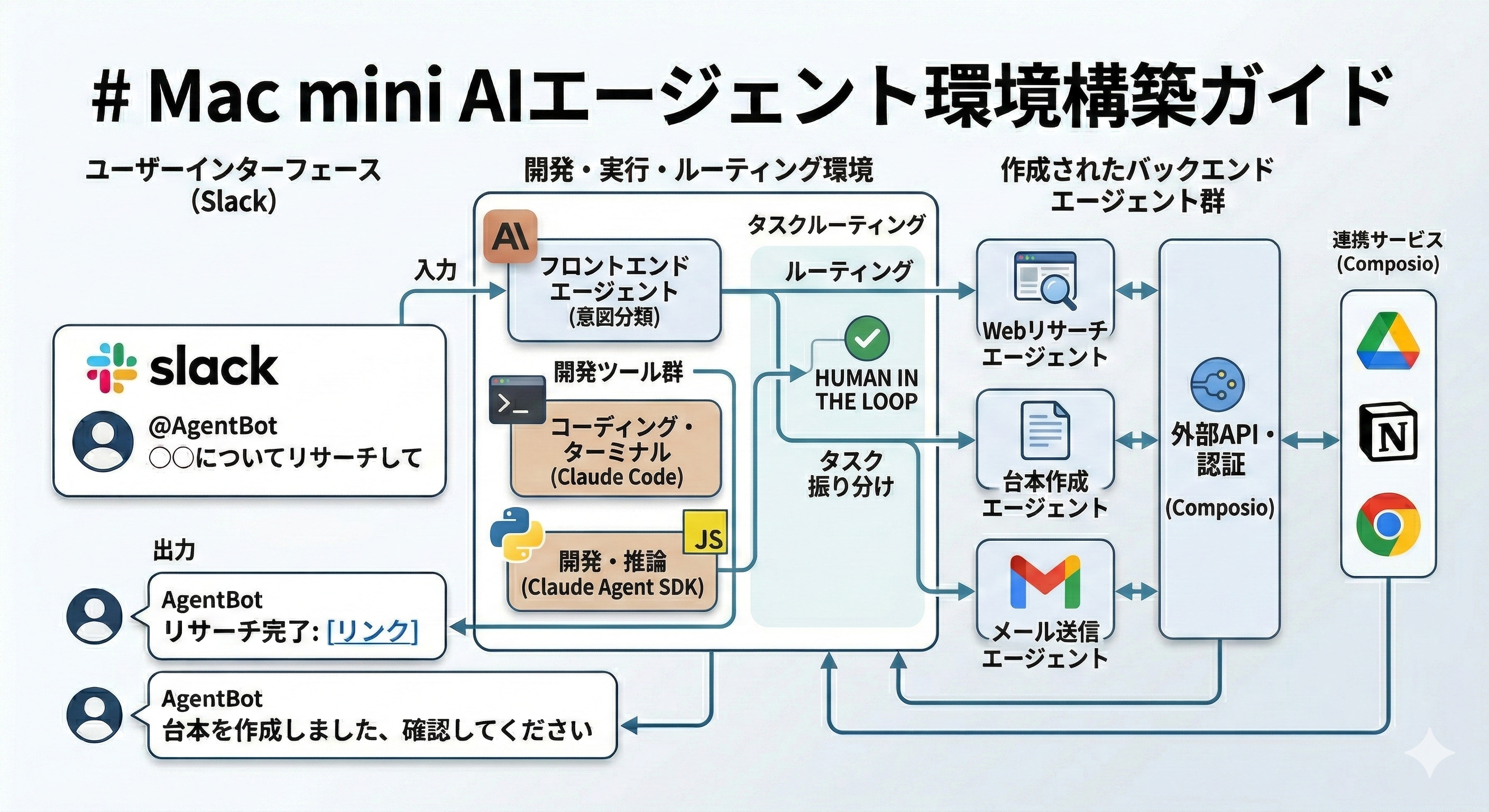
# Mac mini AIエージェント環境構築＆ハンズオンガイド

本ドキュメントは、購入したてのMac miniをベースに、AIエージェント環境をゼロから構築し、実際に手を動かしながらAIコーディングやエージェント開発を学ぶための実践的なガイドです。 

## 0. はじめに：Macの初期設定

購入したてのMacを開いたら、まずは標準のSafariブラウザを使って、今後の作業のベースとなる「Google Chrome」をインストールします。

1. Safariを開き、[Google Chromeのダウンロードページ](https://www.google.com/intl/ja/chrome/)へアクセスします。
2. Chromeをダウンロードしてインストールします。
3. インストール後、Chromeを開き、まずは**「ゲストモード」**または個人のGoogleアカウントでログインして作業を進めます。

## 1. 開発環境のセットアップ

AIエージェントを動かすための基本的なツール群をインストールします。ターミナル（Mac標準アプリの「ターミナル」）を開いて順番に実行してください。

### 1.1 Homebrew のインストール

**Homebrew（ホームブルー）とは？**：Mac用の「アプリやツールをコマンド一つでインストール・管理できる」パッケージマネージャーです。

/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

### 1.2 Node.js のインストール

**Node.jsとは？**：JavaScriptをブラウザ以外の場所（サーバーやPC上）で動かすための仕組みです。Claude Codeなどのツールを動かすために必要です。

brew install node  
node -v # バージョンが表示されればOK

### 1.3 Python と uv のインストール

**Pythonとは？**：AI開発で最もよく使われるプログラミング言語です。 **uvとは？**：Pythonのパッケージ（便利なプログラムの部品）を爆速でインストール・管理するための最新ツールです。

brew install python  
brew install uv  
python3 -v # バージョンが表示されればOK

### 1.4 Git と VSCode のインストール

**Gitとは？**：プログラムの変更履歴を保存・共有するためのシステムです。 **VSCode（Visual Studio Code）とは？**：世界中で最も使われている高機能なコードエディタ（メモ帳の進化版）です。

brew install git  
brew install --cask visual-studio-code

#### VSCode の初期設定

VSCodeを起動し、左側のブロックアイコン（拡張機能）から以下を検索してインストールします。

1. **Japanese Language Pack**: メニューを日本語化します（インストール後、右下の再起動ボタンをクリック）。
2. **Claude Code**: Anthropic公式の拡張機能。エディタ内でAIと対話しながらコーディングできます。
3. **Python**: Pythonコードの補完やエラーチェックに必須です。

## 2. APIキーとモデルの選定

AIをプログラムから動かすには「APIキー（AIを使うためのパスワードのようなもの）」が必要です。用途とコスト（100万トークン＝約75万文字あたりの価格）に応じてモデルを選びます。

### 2.1 Claudeの利用形態と課金について（重要）

Claudeを利用するには、大きく分けて2つの形態があり、**それぞれ課金体系が異なります。**

1. **Claude Pro (サブスクリプション)**
   * 月額課金でブラウザ版（Claude.ai）やClaude Codeを利用するためのプランです。
   * 使用状況の確認: [Claude.ai Usage Settings](https://claude.ai/settings/usage)
2. **Anthropic API (従量課金)**
   * **Claude Agent SDKなど、自作のプログラムからAIを呼び出す場合はこちらが必須です。** サブスクリプションのプランは使えません。
   * 事前にクレジットカードで残高をチャージ（Prepaid）して使用します。
   * 残高の確認・チャージ: [Anthropic Console Billing](https://platform.claude.com/settings/billing)
   * APIキーの作成: [Anthropic Console API Keys](https://platform.claude.com/settings/keys) から「Create Key」をクリックして取得します。

### 2.2 モデル価格比較表（1Mトークンあたり / USD）

| プロバイダー | モデル名 | 入力 (Input) | 出力 (Output) | 特徴・用途 |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Anthropic** | Claude 4.6 Opus | $15.00 | $75.00 | 最高性能。複雑な推論や高度なコーディング向け。 |  |  |
|  | Claude 4.5 Sonnet | $3.00 | $15.00 | **推奨**。性能とコストのバランスが良く、日常的な開発に最適。 |  |  |
|  | Claude 4.5 Haiku | $0.25 | $1.25 | 高速・低コスト。意図分類や簡単なテキスト処理向け。 |  |  |
| **OpenAI** | GPT-4o | $2.50 | $10.00 | Claude Sonnetと同等のバランス型。 |  |  |
|  | GPT-4o-mini | $0.15 | $0.60 | Haikuと同等の低コストモデル。 |  |  |
| **Google** | Gemini 1.5 Pro | $1.25 | $5.00 | 長いコンテキスト（大量の資料読み込み）に強い。 |  |  |
| **DeepSeek** | DeepSeek V3 | $0.14 | $0.28 | 圧倒的な低コスト。単純作業の大量処理に最適。 |  |  |

*参考:* [*Anthropic API Pricing*](https://www.anthropic.com/pricing)

## 3. Claude Code のインストールと使い方

**Claude Codeとは？**：ターミナル上で直接Claudeと対話し、ファイルの作成、コードの編集、コマンドの実行を自律的に行ってくれるAIコーディングアシスタントです。

### 3.1 インストールと認証

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

ブラウザが開くので、Anthropicアカウントでログイン（OAuth認証）します。

### 3.2 基本的な使い方とコマンド

ターミナルで claude と打つと対話モードに入ります。以下のコマンドを覚えておきましょう。

* /model: 使用するAIモデルを変更します（例：コストを抑えたい時はHaikuに変更）。
* /plan: **（重要）** すぐにコードを書かせるのではなく、まずは「どう実装するか」の計画（Plan）をAIに考えさせるモードです。複雑なタスクの前に必ず使いましょう。
* /clear: 会話の履歴をクリアします。

*参考:* [*Claude Code Documentation*](https://docs.anthropic.com/en/docs/agents-and-tools/claude-code/overview)

## 4. Skills と Plugins の導入（ハンズオン）

**Skills（スキル）とは？**：Claude Codeに特定のツール操作や業務知識を教え込む機能です。 **Plugins（プラグイン）とは？**：Claude Codeの機能を拡張するための公式・サードパーティ製のアドオンです。

### 4.1 必須スキルの有効化と管理

Claude Code内で /skills と入力し、標準で用意されているスキルを確認します。

# Claude Code内で実行  
> /skills

ブラウザ操作やファイル検索など、必要なものを有効化します。 また、Anthropic Consoleの [Skills管理ページ](https://platform.claude.com/workspaces/default/skills) から、ワークスペース全体で共有するスキルを管理・作成することも可能です。

### 4.2 Claude Code Plugins の導入

Claude Codeはプラグインを導入することで、さらに強力になります。 *参考:* [*Claude Code Plugins Blog*](https://claude.com/blog/claude-code-plugins)

**代表的なプラグインの例:**

* **GitHub Plugin**: PRの作成やレビューをClaude Codeから直接行えます。
* **Vercel Plugin**: デプロイの管理やログの確認が可能です。

**インストール方法:** ターミナルで以下のコマンドを実行してプラグインを追加します（例としてGitHubプラグイン）。

claude plugin add @anthropic-ai/claude-code-plugin-github

### 4.3 【ハンズオン】カスタムスキルをAIに作らせてみよう

相川さんの業務フロー（スライド生成、SNS投稿など）を自動化するには、独自のスキルを追加する必要があります。 ここでは、**「AIに指示を出して、AI自身にスキルを作らせる」**というAIコーディングを体験します。

**手順:**

1. ターミナルで claude を起動します。
2. 以下のプロンプト（指示文）をコピーして、Claude Codeに貼り付けて実行してください。

**プロンプト:** /plan モードを使って計画を立ててから実装してください。 現在のディレクトリに .claude.json というファイルを作成し、そこに「指定したキーワードでWikipediaを検索し、概要を返す」というカスタムツール（Skill）を定義してください。 実際に検索を行うPythonスクリプト（wiki\_search.py）も作成し、.claude.json からそのスクリプトを呼び出すように設定してください。

1. Claudeが計画を提示してくるので、問題なければ y (Yes) を押して実装させます。
2. 完成したら、Claude Code内で「〇〇についてWikipediaで調べて」と指示し、作成したスキルが動くかテストしてみましょう。

## 5. Composio の導入と連携

**Composioとは？**：AIエージェントが、Gmail、Google Drive、Slackなどの外部サービス（SaaS）を簡単に操作できるようにするための「橋渡し（統合）ツール」です。

### 5.1 インストールとログイン

pip install composio-core  
composio login

ブラウザが開くのでアカウントを作成します。

### 5.2 アプリの連携

相川さんのフロー（台本確認、先方連携など）で必要になるツールを連携します。

# Google Workspace（ドキュメント、スプレッドシート、Gmail等）  
composio add google-workspace  
  
# Slack（通知や確認用）  
composio add slack

ブラウザで認証画面が開くので、アクセス権を許可してください。これでAIがあなたの代わりにGoogleドキュメントを作ったり、Slackにメッセージを送れるようになります。

## 6. Claude Agent SDK のセットアップ

**Claude Agent SDKとは？**：Claude Codeの強力な自律動作機能を、Pythonプログラムの中から呼び出して「独自の業務エージェント」を作るための開発キットです。

### 6.1 プロジェクトの準備

git clone <your-repository-url> claude-agent-sdk  
cd claude-agent-sdk  
  
# uvを使って仮想環境（プロジェクト専用のPython環境）を作成  
uv venv  
source .venv/bin/activate  
  
# 依存関係のインストール  
uv pip install -r requirements.txt

### 6.2 ブラウザ操作ツールの追加

エージェントにWebサイトの画面を操作させるため、Playwright（ブラウザ自動操作ツール）をインストールします。

uv pip install playwright  
playwright install

### 6.3 環境変数（.env）の設定

**.env ファイルとは？**：APIキーなどの「パスワード情報」をプログラムに読み込ませるための隠しファイルです。絶対にGitHubなどで公開しないためにenvにまとめて記入します｡

プロジェクトのルート（一番上の階層）に .env という名前のファイルを作成し、以下のように記述します。

# Anthropic API Key (Claudeを使うための鍵)  
ANTHROPIC\_API\_KEY=sk-ant-api03-...  
  
# Slack Bot Tokens (Slackと連携するための鍵)  
SLACK\_BOT\_TOKEN=xoxb-...  
SLACK\_APP\_TOKEN=xapp-...  
  
# Composio API Key (外部ツールと連携するための鍵)  
COMPOSIO\_API\_KEY=ak\_...

## 7. エージェントの作成と実行（ハンズオン）

最後に、Slackをインターフェースとしたエージェントルーターの仕組みを学び、実際に新しいエージェントを作ってみましょう。

### 7.1 アーキテクチャの理解

slack\_bot/main.py が司令塔（ルーター）です。

1. Slackでメンションされると、安価なモデル（Haiku等）が「ユーザーが何をしたいか」を判断します。
2. 目的の業務（リサーチ、台本作成など）を担当する agents/ フォルダ内の専用エージェントに仕事を振ります。

### 7.2 【ハンズオン】新しいエージェントをAIと一緒に作る

「クラウドワークスの候補者情報をまとめる」エージェントの土台を作ってみましょう。ここでもClaude Codeを使います。

**手順:**

1. ターミナルで claude-agent-sdk ディレクトリに移動し、claude を起動します。
2. 以下のプロンプトを入力します。

**プロンプト:** /plan モードを使用してください。 document/agent-creation-guide.md の規約を読んで理解してください。 その後、agents/recruiting/ ディレクトリを作成し、その中に agent.py を作成してください。 このエージェントは、引数として受け取った「候補者のスキル情報」を元に、面接に進むべきかの簡単な評価文を生成して返すシンプルな関数 evaluate\_candidate(skills: str) を実装してください。 戻り値はガイドの規約に従い、output\_file, elapsed\_seconds, cost\_usd などを含む辞書型にしてください。

1. Claudeが規約を読み込み、正しいフォーマットでPythonファイルを作成してくれます。
2. 次に、このエージェントをSlackから呼び出せるようにします。

**プロンプト:** 今作成した agents/recruiting/agent.py の evaluate\_candidate 関数を、slack\_bot/main.py から呼び出せるようにルーティング処理を追加してください。 ユーザーがSlackで「候補者の評価をして」と言った場合に反応するように、意図分類のプロンプトも修正してください。

### 7.3 エージェントの起動とテスト

すべての実装が終わったら、ターミナルで以下のコマンドを実行してBotを起動します。

# claude-agent-sdk ディレクトリで実行  
uv run python slack\_bot/main.py

Slackを開き、Bot宛に @AgentBot 候補者の評価をして。スキルはPythonとReactです。 とメンションを送ってみましょう。 裏側であなたが作ったエージェントが動き、Slackに返答が返ってくれば成功です！