Ollama と Super-Agent-Partyの探求（Windows上）

https://www.kkaneko.jp/cc/aitasks/super-agent-party.html

金子邦彦

演習の位置づけ

- 本日の作業：**Ollama と Super Agent Party をセットアップ**し、**チャットで動作確認**することに挑戦開始。
- 演習はゴールではなくスタート。**基礎環境を整えた後の試行と探求が本来の目的**。
- 各自で別モデルを試したり、Super Agent Party の各機能（後掲）を実際に触って学んでほしい。
- 学ぶ意義：「AIを使う側」から「**AIを構築・運用・拡張する側**」へ踏み出す第一歩。

ローカルLLM

- **自分のPC内**で**大規模言語モデル（LLM）を動かす**方式。
- API料金不要、オフライン動作が可能、**機密データも自分の管理下で扱える。**
- 応答速度と扱える最大モデルは、PCの GPU（描画演算チップ）とメモリ容量で決まる。

- 学ぶ意義：**用途に応じて「クラウドAI／ローカルAI」を選び分ける**判断力が身に付く。

モデル（演習指定の gemma4:e4b）

- 学習済みのモデルのファイル。**LLMの本体**に相当する。
- 「e4b」の E は「effective（実効）」の略、推論時に使う実効パラメータが約 4B（40億）。PLE（Per-Layer Embeddings＝層ごとの埋め込み）技術で**軽量化**し、一般のパソコンで動作可能。

- Ollama を用いたモデル取得コマンド

ollama pull gemma4:e4b

Ollama（ローカルLLM 実行ランタイム）

- LLMを動かす実行環境
- Ollama の3つの基本コマンドで操作

serve（常駐起動、既定ポート 11434 で HTTP API を公開）

pull（モデル取得）

run（対話）

- 動作確認：
  ollama run gemma4:e4b で対話開始、/bye で終了。
- 学ぶ意義：ローカルLLM運用の手順を習得し、**自前のAI環境構築の基礎**が身に付く。

（参考）Ollama の環境変数（資料で指定された設定値）

- OLLAMA_FLASH_ATTENTION=1：Flash Attention（注意機構の高速化技術）を有効化、長文処理を効率化。
- OLLAMA_KV_CACHE_TYPE=q8_0：KVキャッシュ（生成中の一時記憶）を8bit量子化（数値精度を落として軽量化）、メモリ消費を約半分に（※Flash Attention が前提）。
- OLLAMA_CONTEXT_LENGTH=8192：コンテキスト長（一度に扱えるテキスト量、既定4096）を拡張。
- OLLAMA_MODELS=C:¥Ollama¥models：モデル保存先を指定（既定は %USERPROFILE%¥.ollama¥models）。

Ollama の環境変数（資料で指定された設定

プロバイダー（Provider＝LLMの提供元）

- LLMの提供元を共通の窓口で扱う
- **クラウド型**（OpenAI 等）と**ローカル型**（Ollama 等）に分かれる。
- Super Agent Party では Models ページの Model Providers から Add Provider で追加する。種別に ollama を選び Confirm Add を押すと、ローカルの Ollama ランタイムと接続される。

Super Agent Party（AIエージェント基盤）

- チャット・インタフェースの機能の他、**LLMに知識検索・記憶・ツール実行等を付与しエージェント化（自律的にタスクを進めるAI化）**する機能。
- 公式ページの「Windows Desktop Installation」からインストール、起動後に上記Providerを追加する。
  虫眼鏡ボタンで Ollama 内のモデル一覧を取得→使用モデルを選び、チャット画面で利用する。

学ぶ意義：「AIエージェント」の概念に触れ、応用可能性を体感できる。

Ollama の主な機能一覧（公式ドキュメント docs.ollama.com）

- モデル管理：pull（取得）／ls（一覧）／rm（削除）／ps（実行中確認）／stop（停止）
- 対話実行：run でCLI対話、/bye 終了、/? でコマンドヘルプ表示
- カスタムモデル：Modelfile で独自プロンプト・パラメータを設定し、create で生成
- HTTP API：既定 11434 ポートで提供、OpenAI 互換および Anthropic Messages API 互換あり
- マルチモーダル：Vision（画像認識）対応モデルで画像入力が可能
- Tool calling：LLMが外部ツールを呼び出す機能（対応モデルのみ）構造化出力：JSON 等の決まった形式で応答させる機能
- Thinking：応答前に思考過程を行う「思考モード」（対応モデル）
- mbedding：テキストをベクトルに変換（検索や RAG 用途）
- Web検索：内蔵のWeb検索機能（対応モデルから呼出可能）

Super Agent Party の機能一覧

- マルチプロバイダー対応：OpenAI／Ollama／Dify 等、ローカル・クラウド両方のLLMを統一UIで切替
- マルチモーダル統合：推論・画像認識・画像生成・音声認識／合成（ASR／TTS）の組合せ利用
- VRM デスクトップペット：VRM（3Dアバター形式）で常駐、音声対話、OBS／VMC連携対応
- メッセージングbot：QQ／Feishu／Telegram／Discord／Slack へワンクリック展開
- ライブ配信bot：Bilibili／YouTube／Twitch 連携、360度パノラマ配信対応
- ナレーターbot：長文ナレーション、EPUB（電子書籍形式）の音声化、デジタルヒューマン動画
- チャット画面：数式／mermaid（図表記法）／HTML描画、カプセルモード等
- ロールプレイ：tavern 形式キャラクターカード、キャラ毎の声・アバター、長期記憶
- マルチキャラグループチャット：複数AIキャラとの同時対話
- AI Browser：AIエージェント専用ブラウザによる自律的Web操作
- Task Center：バックグラウンドでのタスク自動実行（MCP／Skills 利用）
- 内蔵ツール群：Web検索、知識ベース（RAG）、スマートホーム制御、ブラウザ操作、コード実行
- カスタムツール：MCP／Skills／A2A／HTTP リクエスト／任意のLLMをツール化
- 拡張プラグイン：公式リストからインストール、自作プラグインの開発・公開も可能