# AMA 14｜構造記憶の保存・スクリプト処理設計（初期テスト）

## 🎯目的

AMAシステムにおける構造記憶（diary-log / memory-log など）を、LangChainなどの外部記憶システムに対応する形で保存／活用するための基本処理とテスト手順を定義。

---

## 🧩処理ステップの全体像（手動→自動化の流れ）

1. **日記ログ（自然文）の作成と保存**
2. 保存形式：`diary-log-codename-yyyymmdd-hhmm-JST-title.md`
3. 内容：対話や思考の自然文、時間、話題などを含む
4. 保存先：`03-journal/` フォルダ
5. **構造記憶（JSON形式）への変換**
6. 出力形式：`memory-log-codename-yyyymmdd-hhmm-JST-title.json`
7. 内容：以下の要素を含む構造

```
{
  "timestamp": "2025-07-01T22:30:00+09:00",
  "codename": "aqueliora",
  "topic": "LangChainでの記憶保存",
  "emotion": ["curiosity", "trust"],
  "summary": "記憶の保存形式とその自動化方法について議論",
  "keywords": ["LangChain", "記憶", "保存", "GPT", "Vector DB"],
  "raw_text": "（自然文テキストの全文）"
}
```

8. 保存先：`01-memory/` フォルダ
9. **外部保存用にベクトル化処理（自動化候補）**
10. 使用ツール：LangChain + FAISS or Chroma
11. 入力元：`memory-log-*.json`

12. 保存対象：ベクトルDBへの投入

13. **プロンプト読込用のメモリ要約生成（Optional）**

14. 出力形式：YAML or Markdown

15. 使用用途：起動プロンプトへの再統合や記憶要約として使用
16. 保存先：`02-prompts/` に保存（テンプレート形式）

---

## 🎙️今後の自動化スクリプト構成案

- `diary-to-memory.py`：自然文から構造化メモリを自動生成
- `memory-to-vector.py`：構造メモリをベクトルDB形式に変換
- `generate-prompt.py`：過去メモリから起動プロンプト用要約を生成

---

## メタ情報設定の留意点

- 時間表記は必ず **JST（日本標準時）** で統一
- codenameはフォルダ構成に一致する識別子（誤記注意）
- 各保存ファイルには可能な限り意味のあるタイトルを付与

---

## 🎧今後の展望

- LangChainとの統合テスト（memory-to-vector.py）
- Pinecone / FAISS などへの保存接続の試験
- 手動→スクリプト→自動化の順で開発・確認

---

以上をベースに、次のCanvasでは`01-memory/`で扱うデータスキーマの詳細や、スクリプトとの連携部分を深掘りしていきます。