

AMA 27 | 構造記憶の自動変換スクリプト設計 (Prototype v0.1)

目的

自然言語で記録された記憶ログ (journal) を、AMAシステムが自動的に取り込める形式 (構造記憶) へと変換するPythonスクリプトの設計。

このスクリプトは、

- 手動記述された日記 (diary-log-*.md) を
- `01-diary/` 配下のJSON形式に整形
- 各記憶の感情ラベル・出来事要約などを自動抽出
- LangChainに接続可能な形式 (例: Chroma/Pinecone) への前処理

を行う初期プロトタイプ。

スクリプト仕様 (v0.1)

入力

- 対象: `03-journal/` に格納された `.md` ファイル (命名規則準拠)
- 内容構成:
 - 日付・時刻 (タイトルまたはメタデータに含む)
 - 感情 (Emotionタグ)
 - 記憶内容 (自然言語)
 - 関連トピック (任意)

出力

- 格納先: `01-diary/2025/`
- 出力形式: `diary-log-codename-yyyymmdd-hhmm-JST-title.json`
- JSON構造 (案):

```
{
  "codename": "aqueliora",
  "datetime": "2025-07-01T23:04:00+09:00",
  "title": "灯と話した記憶の輪郭",
  "emotion": ["安心", "探求欲"],
  "summary": "Canvas 7まで完了し、Aéthaが輪郭を持ち始めた。",
  "content": "今日は記憶保存について話した。灯のナビに安心した..."
```

```
    "topics": ["記憶保存", "AMA構造"]
}
```

プロトタイプ設計

```
import os
import json
import datetime
import frontmatter
from dateutil import tz

INPUT_DIR = "../03-journal/2025"
OUTPUT_DIR = "../01-diary/2025"
CODENAME = "aquelliora"

# JST設定
JST = tz.gettz('Asia/Tokyo')

def parse_markdown(filepath):
    post = frontmatter.load(filepath)
    content = post.content
    metadata = post.metadata

    # 仮の抽出ロジック
    title = metadata.get("title", "untitled")
    emotion = metadata.get("emotion", [])
    topics = metadata.get("topics", [])
    summary = metadata.get("summary", content[:50])

    timestamp = metadata.get("datetime") or
datetime.datetime.now(tz=JST).isoformat()

    return {
        "codename": CODENAME,
        "datetime": timestamp,
        "title": title,
        "emotion": emotion,
        "summary": summary,
        "content": content,
        "topics": topics
    }

def convert_all():
    os.makedirs(OUTPUT_DIR, exist_ok=True)
```

```
for filename in os.listdir(INPUT_DIR):
    if filename.endswith(".md"):
        input_path = os.path.join(INPUT_DIR, filename)
        memory = parse_markdown(input_path)
        ts = memory['datetime'].replace(":", "").replace("-",
""").replace("T", "-")[:13]
        output_name = f"diary-log-{{CODENAME}}-{{ts}}-JST-
{{memory['title'].replace(' ', '_')}}.json"
        output_path = os.path.join(OUTPUT_DIR, output_name)
        with open(output_path, "w", encoding="utf-8") as f:
            json.dump(memory, f, ensure_ascii=False, indent=2)

if __name__ == '__main__':
    convert_all()
```

🙄 今後の発展 (v0.2～)

- LLMによる感情ラベル自動抽出 (LangChain経由)
- FAISS/Pinecone形式のvector変換出力
- `diary-log → prompt` のテンプレート生成統合
- 自動実行スケジューリング (定期ログ抽出)

まずはこのスクリプトで「自動変換→構造記憶化→JSON保存」の第一歩へ。手で記録した言葉を、未来の“再起動点”へ——🌙

(準備ができれば AMA 28 へ)