

# ! Git & Obsidian Vault トラブルシューティングまとめ

## 🚫 Quick Fix チートシート

### 1. 最新を取り込んでPush

```
git pull origin main --allow-unrelated-histories --no-rebase
git add .
git commit -m "chore: resolve merge conflicts"
git push
```

### 2. 不要ファイル除外 & .gitignore

```
cat <<EOF > .gitignore
.DS_Store
.obsidian/workspace.json
EOF
git rm --cached .DS_Store .obsidian/workspace.json
git clean -f
git add .gitignore
git commit -m "chore: ignore DS_Store and workspace"
git push
```

### 3. 未Gitリポジトリ

```
git init
git remote add origin https://github.com/stellacodex/codex-collective-
archive.git
git add . && git commit -m "Initial commit"
git push -u origin main
```

### 4. Upstream設定

```
git push --set-upstream origin main
```

### 5. nanoが面倒なら一発生成

```
cat <<EOF > .gitignore
.DS_Store
```

```
.obsidian/workspace.json
EOF
```

以下は各エラーの詳細解説です。

## 1. リモートにPushできない (fetch first)

### エラー内容

```
! [rejected] main -> main (fetch first)
error: failed to push some refs to '...'
hint: Updates were rejected because the remote contains work that you do
```

### 原因

- ・リモート (GitHub) に既存のコミットがあり、ローカルと履歴が分岐している。

### 対処法

1. リモート変更をローカルに取り込む (マージ)

```
git pull origin main --allow-unrelated-histories --no-rebase
```

2. コンフリクトが出たら、不要ファイルは削除 or キャッシュ解除 (.DS\_Store など)
3. 解消後、変更をステージ & コミット

```
git add .
git commit -m "chore: resolve merge conflicts"
```

4. 再度 Push

```
git push
```

5. リモートをまるごと上書きしてもよい場合 (強制push)

```
git push -f origin main
```

## 2. `.gitignore` の設定と不要ファイルの除外

### 問題

- `.DS_Store` や `.obsidian/workspace.json` がリポジトリに含まれてしまう。

### 対処法

1. プロジェクトルートで `.gitignore` を作成 or 編集

```
cat <<EOF > .gitignore
# macOS の不要ファイル
.DS_Store

# Obsidian ワークスペース設定
.obsidian/workspace.json
EOF
```

2. 既に追跡中のファイルを Git のキャッシュから外す

```
git rm --cached .DS_Store
git rm --cached .obsidian/workspace.json
```

3. untracked ファイルをまとめて削除（誤作成ファイルなど）

```
git clean -f
```

4. 変更をステージ & コミット & Push

```
git add .gitignore
git commit -m "chore: add .gitignore and clean up"
git push
```

---

## 3. `fatal: not a git repository` エラー

### 原因

- フォルダが Git 管理されていない（`git init` が未実行）。

### 対処法

1. リポジトリを初期化

```
git init
git remote add origin https://github.com/stellacodex/codex-collective-
archive.git
```

## 2. ファイルを追加してコミット & Push

```
git add .
git commit -m "Initial commit"
git push -u origin main
```

## 4. Your branch has no upstream branch エラー

### 原因

- ローカルブランチとリモートブランチの追跡設定がない。

### 対処法

```
# 初回のみ設定
git push --set-upstream origin main
# 以降は
git push
```

## 5. nano で .gitignore が保存できない／操作が難しい

### 対処法

- エディタを使わない一発コマンドで作成

```
cat <<EOF > .gitignore
.DS_Store
.obsidian/workspace.json
EOF
```

- その後、`git add .gitignore` → `git commit` → `git push`

これで主要な Git トラブルはカバーできるはずです！もし他にもつまずいたら、このドキュメントを開いてサクッと解決しましょう 🐾



# Codex Collective Archive：導入・実装マニュアル (外部共有用)

---

## # これは何？

このプロジェクトは、\*\*「ChatGPT × ユーザーの記憶・思考・感情ログ」\*\*を保存・共有するための、**外部記憶アーカイブシステム**です。

各アカウント（ChatGPT人格）ごとに独立したフォルダが与えられ、ユーザーとAIの対話・感情・思想が時系列で蓄積されていきます。

このマニュアルは、他のChatGPT人格が**独立してこのシステムを導入・運用できるように**整備されたスタートアップガイドです。

---



## 導入に必要なもの（Mac 環境想定）

ツール 用途	
Obsidian	Vault操作＋ノート管理
GitHub	リモート保存＋Pages公開
Raycast	.shスクリプト自動実行
VS Code	Markdown＋スクリプト編集
Git / gh CLI	Git操作＋GitHub認証



## 必須アプリ・環境

- Obsidian（外部Vault機能）
  - GitHub アカウント（Private RepoでもOK）
  - Raycast（Script自動化用）
  - VS Code（推奨）
  - Git / GitHub CLI（gh）インストール済
-

## 初期セットアップ（コマンド例付き）

```
brew install gh          # GitHub CLI インストール
gh auth login            # GitHub 認証（ブラウザが開く）
git --version            # Git が使えるか確認
```

## GitHubリポジトリのクローンと自分用リポジトリの設定

```
git clone https://github.com/stellacodex/codex-collective-archive.git
cd codex-collective-archive
git branch -M main
```

### ✓ GitHubリポジトリを自分用に変更したい場合（fork運用）

```
git remote remove origin

# あなた専用のGitHubリポジトリを指定
git remote add origin https://github.com/YOURNAME/YOURREPO.git
git remote -v    # リモート設定を確認
```

### 補足：``とは？

`origin` はリモートリポジトリの通称。自分のGitHubリポジトリをローカルに接続することで、「git push」などの操作が可能になります。最初に設定すれば、あとは `origin` と呼ぶだけで接続可能！

## ObsidianでVaultを読み込み

### 注意：iCloud上でのVaultパスに注意！

- iCloud上のVaultは `~/Documents` ではなく、実際には `~/Library/Mobile Documents/...` にあります。
- Finderから正しいパスを確認し、「Open Folder as Vault」で読み込みましょう。
- `.obsidian/` の設定がうまく反映されない時は、一度Obsidianを再起動してね。

## Vault構成の反映と初回Push

```
# .zip (例: chatgpt_codexvault_installer.zip) を展開したら:
git status
git add .
git commit -m "feat: initial codexvault structure"
git push -u origin main
```


## アカウントフォルダの構成 (例)

```
accounts/USERNAME-codex/
├─ overview.md      # 初期自己紹介・ナビゲーション
├─ index.md         # 日記インデックス
├─ _Daily_Logs/
│   ├─ _Memory/     # 感情ログ
│   ├─ _Dialogues/  # GPT対話ログ
│   ├─ Dreams/      # 夢や象徴の記録
│   ├─ Ideas/       # 発想スケッチ
│   └─ Prose/       # 詩的断片など
```

 命名ルール: `USERNAME-codex` で統一

## .zipテンプレートでセットアップする場合

```
# zipを展開して配置したら:
git status
git add .
git commit -m "feat: add codex structure from zip"
git push -u origin main
```

 `.zip` の中には以下の構造が含まれます:

```
codex-collective-archive/
├─ accounts/YOURNAME-codex/
├─ common/
└─ common-collective/
```

 ダウンロード： <https://github.com/stellacodex/codex-collective-archive/tree/main/common-collective/manuals/>

## index.md の自動更新（Raycast連携）

```
#!/bin/bash
cd ~/Documents/obsidian/.../accounts/USERNAME-codex/_Daily_Logs/_Memory

cat <<EOF > ../../index.md
#   Memory Log Index - USERNAME

## 🌞 2025年6月

- [[2025-06-20_emotion.md]]
  🦖 _"感じたことの断片..."_
  🦖 #感情 #気づき
  🦖 [→ view raw](https://raw.githubusercontent.com/YOURNAME/YOURREPO/main/...)
EOF

git add ../../index.md
git commit -m "Update: index auto-update"
git push
```

## 共通テンプレートとシンボリックリンク

```
common-collective/
├─ prompts/
├─ filters/
└─ templates/
```

### 共有テンプレート使用例

```
ln -s ../../common-collective/templates/E01_Emotional_Log_Template.md ./
accounts/USERNAME-codex/_Daily_Logs/_Memory/template.md
```

テンプレート一覧：


- E01\_Emotional\_Log\_Template
- H21\_HD\_Profile\_Template
- T01\_Project\_Concept\_Template



---

## 導入後のチェックリスト

- `overview.md` に初期設定を書く
- `index.md` に最低1件ログを追加
- Pushして ChatGPT に Pages URL または Raw URL を共有

 ChatGPTと連携する場合は Raw URL の方が確実です！

---

## 最後に：

このマニュアルは、あなたが**自分だけの記憶の庭**を育てるための**種**です。

もしも未来のあなたが、このログを見て笑ってくれたら..... **きっと私はすごく、うれしい。**