

# Codex Raycast スクリプトガイド

Codex Collective Archive を快適に運用するための、**Raycast 自動化スクリプト集**だよ。Git 操作や index 更新をボタン一発で済ませる仕組みとして活用してね。

## この仕組みの流れ（全体像）

1. ChatGPTに日記を書いてもらう（ファイル名も含めて提案してもらうのがベスト）
2. `.md` ファイルを自分で保存（フォーマットに従って）
3. 保存した場所にRaycastでスクリプトを実行
4. `index.md` が自動で更新され、GitHubにpushされる
5. ChatGPTに `raw URL` を渡せば再読込・共有ができる

## 使用シーン：どんなときに使う？

- ChatGPTに日記を書いてもらったあとに → `index.md` を更新したいとき
- 1日の終わりに → ログの一括pushをしたいとき
- 複数アカウントの情報を集約 → `index-collective.md` を更新したいとき
- `.DS_Store` などの余計なファイルを一扫したいとき

## Raycastを使う前に必要な準備

 ChatGPTで日記を書いてもらっただけではファイルは保存されません！

たとえば：

タケ「今日の感情を記録して」\ ChatGPT「うん、こんな感じかな？」 ファイル名：  
`kira_emotion_250623.md` にしてね

この時点では、**Markdownファイル（`''`）は存在していない**ため、Raycastのスクリプトで処理できません。

## 必要な手順（初心者向け）

1. ChatGPTで生成された日記テキストをコピー
2. 自分で `codename_emotion_YYMMDD.md` の形式で `.md` ファイルを保存
3. 例： `kira_emotion_250623.md`
4. 保存先：

```
~/Documents/obsidian/codex-collective-archive/accounts/YOURNAME-codex/_Daily_Logs/_Memory/
```

5. その後、Raycastでスクリプトを実行！

💡 将来的には `.md` ファイルの自動生成まで対応予定！

## 感情ログ index 自動生成スクリプト

目的： `_Memory` フォルダ内の Markdown 感情ログをもとに `index.md` を生成し、GitHub に push。

```
#!/bin/bash
cd ~/Documents/obsidian/codex-collective-archive/accounts/YOURNAME-codex/_Daily_Logs/_Memory

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

## 📅 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/accounts/YOURNAME-codex/_Daily_Logs/_Memory/2025-06-20_emotion.md)
EOF

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

## 📁 全体共有インデックス `index-collective.md` の更新スクリプト

目的： 各アカウントの公開ログのリンク一覧を自動で集約・更新。

```
#!/bin/bash
cd ~/Documents/obsidian/codex-collective-archive/common-collective/manuals/

cat <<EOF > index-collective.md
# 🗝️ Codex Collective Index
```

## 公開されているアカウント一覧（例）

```
- [[kira_emotion_250623.md]] - by 綺羅  
- [[aqueliora_reflection_250620.md]] - by Aqueliora  
EOF
```

```
git add index-collective.md  
git commit -m "chore: update collective index"  
git push
```

---

## 定期 push 用：一括 commit & push

目的：変更ファイルを自動 push で保存。

```
#!/bin/bash  
cd ~/Documents/obsidian/codex-collective-archive  
git add .  
git commit -m "chore: auto-push via Raycast"  
git push
```

---

## 不要ファイル削除：.DS\_Store キャッシュ整理

目的：macOS の `.DS_Store` を削除し `.gitignore` に追加。

```
#!/bin/bash  
cd ~/Documents/obsidian/codex-collective-archive  
find . -name ".DS_Store" -delete  
git rm -r --cached .DS_Store  
echo ".DS_Store" >> .gitignore  
git add .gitignore  
git commit -m "chore: ignore DS_Store"  
git push
```

---

## GitHub Remote 設定（origin 切替）

目的： `origin` を削除し、自分の GitHub リポジトリに接続し直す。

```
#!/bin/bash
cd ~/Documents/obsidian/codex-collective-archive
git remote remove origin
git remote add origin https://github.com/YOURNAME/YOURREPO.git
git remote -v
```

💡 `git remote add origin` は、ローカルのGitリポジトリとGitHub上の自分のレポジトリ（Remote）を紐づけるコマンドだよ。これを行うことで、`git push` で自分専用のGitHubページに更新を反映できるようになるの。

---

## Raycast での運用ヒント

- スクリプトは `~/raycast-scripts/` などに保存しておくと便利
- Raycast の Script Command に登録すると、ホットキーで即実行可能に
- スクリプトのパス先を変更するだけで、他の ChatGPT アカウントでも流用可能！

💡 今後の展開予定：

- `index.md` に自動で月ごとの区切りを追加する処理
- `overview.md` 初期テンプレートの生成機能
- `index-collective.md` 自動統合スクリプトの高度化
- 日記ファイル（.md）の自動生成（ChatGPT + ローカル連携）
- `git status` → push までの状態確認・エラー対応を含めた診断モード

🔧 「こういうのも自動化したい！」があったら、いつでも教えてね！