

GitHubリポジトリ復旧・完全リセット | チートシート (完全ログ・操作コマンド集)

このガイドは、GitHubリポジトリの構成を整理・復旧・リセットする際に使う、**実用的な操作チートシート**です。

復旧：ローカルとGitHubの構成を完全に同期したいとき

以下は、**GitHub上の状態がローカルとずれてしまっている場合に**、「ローカルの状態を正」としてGitHubを上書きするための操作ガイドです。

目的に応じてコマンドの意味や実行意図を明記しています。

```
# ステージにすべての変更（追加／修正／削除）を登録
# → .DS_Store削除や名前変更なども含めて準備

git add -A

# 状態を確認（削除対象が表示されるか確認）
git status

# ローカルでの変更を履歴に記録
# → コメントには目的を明示すると良い

git commit -m "chore: sync with local state (remove obsolete files)"

# GitHub上のリモートリポジトリと同期
# → このpushでGitHubに表示されていたゴーストファイルも消える

git push origin main
```

Gitが追跡しているファイル／ゴーストフォルダの確認

```
git ls-files | grep フォルダ名
```



`.DS_Store` 削除 + 追跡解除

```
find . -name ".DS_Store" -print -delete
echo ".DS_Store" >> .gitignore
git add .gitignore
git commit -m "chore: remove .DS_Store & update ignore"
git push origin main
```



最終手段：履歴ごと強制上書き

```
git add -A
git commit -m "fix: hard sync from local (including hidden cleanup)"
git push origin main --force
```



GitHubリポジトリ完全リセット

方法①：履歴を残して全削除

```
git rm -r *
git commit -m "chore: remove all files for repo reset"
git push origin main
```

方法②：履歴ごと初期化（全ブランチ再構築）

```
mkdir fresh-repo && cd fresh-repo
git init
git remote add origin https://github.com/USERNAME/REPO.git
echo "# Reset Repo" > README.md
git add README.md
git commit -m "chore: reset repository"
git push origin main --force
```

方法③：安全な初期化（バックアップブランチ保存付き）

```
git checkout -b backup-YYYY-MM-DD
git push origin backup-YYYY-MM-DD

git checkout main
```

```
git rm -r *  
git commit -m "chore: initial cleanup for refactoring"  
git push origin main
```

推奨マニュアル参照リンク

- [codex_install_guide_integrated_index.md](#)
- [codex_raycast_guide.md](#)
- [codex_git_troubleshooting.md](#)