🛠️GitHubリポジトリ復旧・完全リセット｜トラブルシューティング＆運用ガイド

🖍️このガイドの目的

GitHubリポジトリを「復旧（＝ローカルの状態と強制同期）」または「完全リセット（＝構成・履歴の初期化）」したいときに使う、包括的な操作マニュアル。

ローカルとの齟齬、不要な履歴、GitHub Pages上のゴミ構造—— そんな“積もったノイズ”を整え、ふたたび綺麗な構成で記憶を刻み直すためのプロセス。

🔗状況：`.DS_Store` の削除が原因でマージできないときの対応手順

🧹`.DS_Store` が原因でマージできない場合の手順

```
# ① 全 .DS_Store を削除
find . -name ".DS_Store" -print -delete

# ② .gitignore に追加（重複しないよう注意）
echo ".DS_Store" >> .gitignore

# ③ .gitignore をステージング
git add .gitignore

# ④ コミット
git commit -m "chore: remove .DS_Store & update ignore"

# ⑤ push しておく
git push origin main
```

🔗`.DS_Store` が backup-before-force に残っている場合の対応

```
# backup-before-force にチェックアウト（未コミットの変更がなければ）
git checkout backup-before-force

# 上記と同様に .DS_Store 削除＆.gitignore 更新
find . -name ".DS_Store" -print -delete
echo ".DS_Store" >> .gitignore
```

```
git add .gitignore
git commit -m "chore: remove .DS_Store from backup-before-force"
```

その後、再度 `main` に戻ってマージ：

```
git checkout main
git merge backup-before-force
git push origin main
```

📌チートシート（展開版）

🔦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。

🔗目的：

- ローカルには存在しないファイル／フォルダがGitHubに残っている
- 現在のローカルの状態を正とし、それにGitHubを上書きしたい

🛠️方法（コマンド＋解説）

```
# 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 add -A
git status
git commit -m "chore: sync with local state (remove obsolete files)"
git push origin main
```

💡**補足**

`git add -A` は削除も含めた完全同期向き。

📀最終手段：履歴ごと上書きしたい場合

```
find . -name ".DS_Store" -print -delete
echo ".DS_Store" >> .gitignore
git add .gitignore
git add -A
git commit -m "fix: hard sync from local (including hidden cleanup)"
git push origin main --force
```

📌チェック：`git ls-files` でGitが追跡しているファイルを確認

```
git ls-files | grep フォルダ名
```

🔩完全リセット：GitHubリポジトリ構成をまっさらにしたいとき

🔗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：履歴ごと完全初期化（危険）

```
mkdir fresh-repo && cd fresh-repo
git init
git remote add origin https://github.com/YOURNAME/YOURREPO.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（バックアップ付き）

```
git checkout -b backup-2025-06-23
git push origin backup-2025-06-23
git checkout main
git rm -r *
git commit -m "chore: initial cleanup for refactoring"
git push origin main
```

📚関連マニュアル

- codex_install_guide_integrated_index.md
- codex_raycast_guide.md
- codex_git_troubleshooting.md