

第3回 Git/GitHubワークフロー × Claude Code

Phase 1 | Git連携 / 講義2h + ワークショップ6h

この回のねらい

- Issue駆動開発の土台 **ブランチ運用** と **Pull Request** を理解する
- Claude Codeに **branch** → **修正** → **commit** → **push** → **PR作成** を一気通貫でやらせる
- **コンフリクト**の発生と解消を体験する

なぜブランチを切るのが

- mainを壊さない
- レビューを挟める
- 並列開発ができる

GitHubの基本フロー

`git switch -c feature/xxx` → 修正 → `commit` → `push`
→ **Pull Request** → レビュー → **merge**

良いコミット / PR

- **コミット**：何を・なぜ（1行サマリ+本文）
- **PR**：変更概要・背景・**テスト結果**・**確認手順**

AIに書かせる前提でも、「何を書くべきか」は人が知っておく。

Claude Codeに任せる際の注意

- `git push --force` / main直push / 大量削除は **人が必ず確認**
- PR作成は **GitHub CLI (gh)** 経由が安定
 - `gh auth login` → `gh pr create`

ワークショップ (6h)

1. ブランチを切り、Claude Codeに修正させる
2. `commit` → `push` → **PR作成** (説明文に確認手順)
3. diffを自分でレビュー → マージ
4. 意図的にコンフリクトを起こして解消
5. AIの差分をレビューア一目線で読む

回のゴール

ブランチを切り、修正し、PRを作成するまでを
Claude Codeで一気通貫できる。

OJT課題：実務で1件、ブランチ→修正→PR作成までを実施