

# 第5回

## 拡張：サブエージェント / MCP / カスタムコマンド

Phase 2 | 環境整備 / 講義2h + ワークショップ6h

## この回のねらい

- Claude Codeを **外部サービス連携（MCP）** と **専門タスクの分業（サブエージェント）** で拡張する
- 定型作業を **カスタムスラッシュコマンド** 化して時短する
- **人＝マネージャー、AI＝実装担当** の役割分担を体感する

## サブエージェント

1つの大きなタスクを **専門役割** に分けて任せる。

- 「調査役」「実装役」「レビュー役」
- 並列で動かす発想（→ 第6回の並列開発につながる）

# MCP (Model Context Protocol)

Claude Codeを **外部サービスと直接つなぐ標準**。

- **GitHub** (Issue / PR) , Slack, データベース 等
- GitHub MCP = IssueやPRをClaude Codeから直接読み書き  
→ **第7・8回のIssue駆動開発の土台**

## カスタムコマンド / 役割分担

- よく使う指示を `/コマンド` 化（例： `/review`, `/test-summary`）
- 人は「計画承認・成果レビュー」、AIは「実装」
- セキュリティ：連携先トークンの権限は最小限に

## ワークショップ (6h)

1. **GitHub MCP** を接続 → Issue一覧・方針提案・コメント投稿
2. 定型作業を **カスタムコマンド化** (例: テスト→結果要約)
3. **サブエージェント** で「実装役/レビュー役」を分けて1タスク処理

# 回のゴール

MCP・サブエージェント・カスタムコマンドで  
**Claude Code**を実務向けに拡張 できる。

**OJT課題**：実務でMCP/カスタムコマンドを使った作業を3件記録