



第12回 「Claude Code 101」
Anthropic 公式 Claude Academy

Claude Code

自動で差し込む

公式の教材を、順番に

原石工房

隠れた才能に気づき、花開く

この資料について



原石工房

Anthropic とは関係のない、非公式の資料です

公式の学習サイト Claude Academy の日本語訳です

原文（英語）Claude Code 101 / Hooks

原文の流れと節の並びをそのままに、要点を落とさず日本語にしています

全51回の、今日はここ



原石工房

Claude Code 101

1 2 3 4 5 6 7 8 9 10 11 12

第1回～第12回

Claude Code in Action

13 14 15 16 17 18 19 20 21

第13回～第21回

スキル

22 23 24 25 26 27

第22回～第27回

手分け (subagents)

28 29 30 31

第28回～第31回

MCP 入門

32 33 34 35 36 37 38 39 40

第32回～第40回

MCP 応用

41 42 43 44 45 46 47 48 49 50 51

第41回～第51回

今日はここ 第12回

今日のレッスン



原石工房

Claude Code 101 の12こめ。ここでこのコースは終わりです

原題 Hooks

訳 自動で差し込む

5つの節と、最後にまとめがあります

このレッスンの流れ



原石工房

教材の並びどおり。この番号が、このあとの見出しに出てきます

- ① なぜ使うのか
- ② どうやって動くのか
- ③ 実際の例
- ④ 道具を使う前に止める
- ⑤ チームで分け合う

① なぜ使うのか



原石工房

原題 Why Use Hooks

「たいてい」と「必ず」の違い

CLAUDE.md に書く

読んでくれる。たいていは、やってくれる

読み飛ばすこともある

フックにする

決まった場面で、毎回そのまま動く

例外はない

毎回、必ず起きてほしいことは、頼みごとではなく仕掛けに書く

頼むのではなく、仕掛けておく

ほかの仕組みは、やってくれることも、やってくれないこともあります

フックは、決まった場面で必ず動きます

CLAUDE.md に書いても、読み飛ばされることはあります

① たいてい動く、と、必ず動く



原石工房

仕掛ける場所 settings.json

① どの場面で

② どの道具のとき

③ 何をさせる

道具を使う前 (PreToolUse)

通す (0) / 止める (2) を返す

終わったあと (PostToolUse)

見た目を整える・記録を残す

返事を終えたとき (Stop)

知らせを出す

止められるのは、道具を使う前だけ

① 何に使うか



原石工房

「たいてい」と「必ず」の違い

CLAUDE.md に書く

読んでくれる。たいていは、やってくれる

読み飛ばすこともある

フックにする

決まった場面で、毎回そのまま動く

例外はない

毎回、必ず起きてほしいことは、頼みごとではなく仕掛けに書く

頼むのではなく、仕掛けておく

書き換えたあとに、見た目を自動で整える

動かした命令を記録に残す

危ないことを止める

仕事を終えたときに、知らせを出す

② どうやって動くのか



原石工房

原題 How They Work

設定のファイルに、3つ決めて書く

どの場面で

下の5つから選ぶ

どの道具のとき

絞り込み（なくてもよい）

何をさせる

動かす命令

場面は、たとえばこの5つ

- PreToolUse 道具を使う前
- PostToolUse 道具の仕事が終わったあと
- UserPromptSubmit こちらが文を送ったとき
- Stop 返事を終えたとき
- Notification 知らせを出すとき

書き方は2つ。「/hooks」と打つか、設定のファイルを自分で書き換えるか

書く場所は、設定のファイル
settings.json です

決めるのは3つ。どの場面で・どの
道具のとき・何をさせる

書き方は「/hooks」と打つか、設定
のファイルを自分で書き換えるか

場面を選び、条件を絞り、させることを書く

② 場面は、たとえばこの5つ



原石工房

PreToolUse 道具を使う前

PostToolUse 道具の仕事が終わったあと

UserPromptSubmit こちらが文を送ったとき

Stop 返事を終えたとき

Notification 知らせを出すとき

設定のファイルに、3つ決めて書く

どの場面で

下の5つから選ぶ

どの道具のとき

絞り込み（なくてもよい）

何をさせる

動かす命令

場面は、たとえばこの5つ

- PreToolUse 道具を使う前
- PostToolUse 道具の仕事が終わったあと
- UserPromptSubmit こちらが文を送ったとき
- Stop 返事を終えたとき
- Notification 知らせを出すとき

書き方は2つ。「/hooks」と打つか、設定のファイルを自分で書き換えるか

場面を選び、条件を絞り、させることを書く

③ 実際の例

原題 A Practical Example

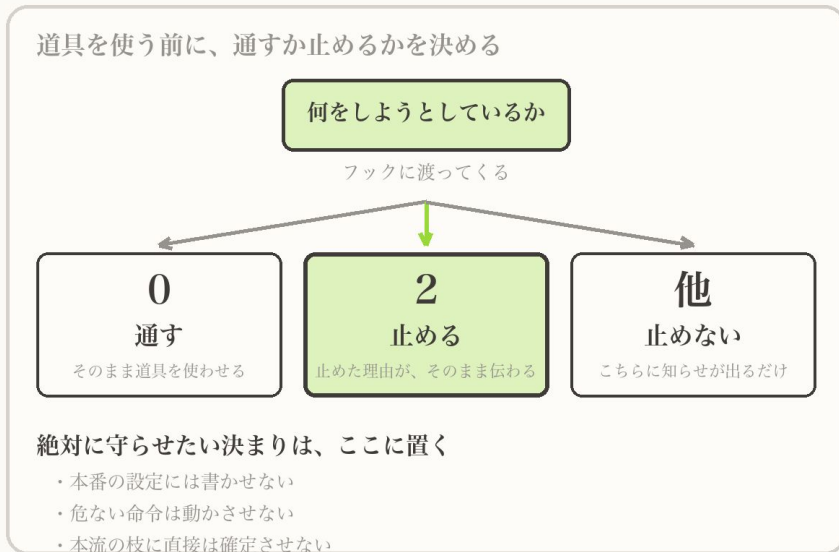
場面は PostToolUse (道具の仕事が終わったあと)

絞り込みは「Edit|MultiEdit|Write」。ファイルに手を入れたときは、いつでも動きます

命令の中でファイルの種類を見て、それに合う道具を当てます

④ 道具を使う前に止める

原題 Blocking with PreToolUse



止める理由も、そのまま伝わる

何をしようとしているかが、フックに渡ってきます

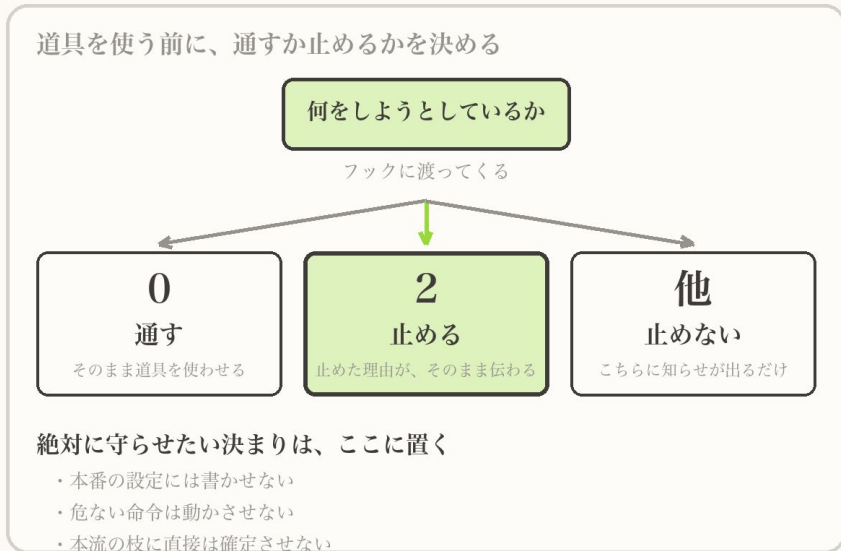
0 を返せば通す。2 を返せば止める

止めたときは、書いた文がそのまま Claude に伝わります

④ 絶対に守らせたい決まり



原石工房



止める理由も、そのまま伝わる

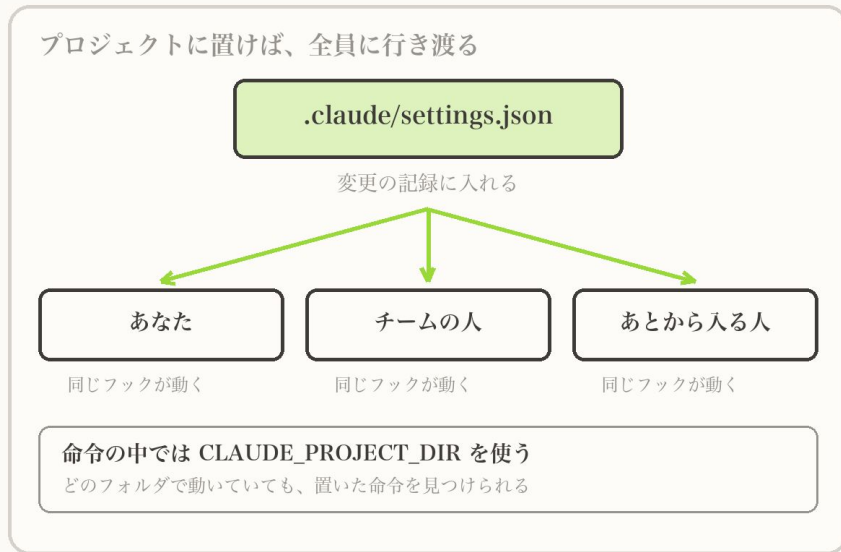
本番の設定が置いてあるところに、書かせない

危ない命令を動かさせない

本流の枝に、直接は確定させない

⑤ チームで分け合う

原題 Sharing Hooks with Your Team



配るのではなく、持ってくれば付いてくる

`.claude/settings.json` に書くと、そのプロジェクトのものになります

変更の記録に入れば、持ってきた人全員に同じフックが行き渡ります

命令の中では
`CLAUDE_PROJECT_DIR` を使って、プロジェクトの中の命令を指します

まとめ



原石工房

原題 Recap

「たいてい」と「必ず」の違い

CLAUDE.md に書く

読んでくれる。たいていは、やってくれる

読み飛ばすこともある

フックにする

決まった場面で、毎回そのまま動く

例外はない

毎回、必ず起きてほしいことは、頼みごとではなく仕掛けに書く

頼むのではなく、仕掛けておく

PostToolUse で、見た目を整える・
記録を残す

PreToolUse で、危ないことを止め
る

書き方は「/hooks」か、設定のファ
イル

今回のまとめ



原石工房

フックは、決まった場面で必ず動く

決めるのは、場面・絞り込み・させることの3つ

止められるのは、道具を使う前だけ

2を返すと止まり、理由がそのまま伝わる

プロジェクトに置けば、チーム全員に行き渡る

次回



原石工房

Claude Code 101 は、ここまでです

次のコースは Claude Code in Action

原題 Steering long sessions

訳 長い作業の舵を取る

全51回の、13回目です

あなたなら、何を必ず動かしますか



原石工房

隠れた才能に気づき、花開く