

研修で作ったハーネスの置き場

置き場	何を書く	研修で作ったもの	注意
.github/copilot-instructions.md	会話のたびに読まれる書き方の指定	Day1 で作成。Day2 で ファイルを分割しない の行を2行に書き換え	20行前後。前提が変わったら直す
AGENTS.md (tpro-workの直下)	プロジェクトの前提。構成、動かし方、触ってよい範囲	Day1 の発展で作成	事実だけを書く。copilot-instructions.md と同じ中身を写さない
.github/instructions/tests.instructions.md	applyTo に合うファイルを扱うときだけ効く決まり	Day3 演習8。期待値は testcases.md から取る	先頭の applyTo: tests/** が無いと全部の会話で読まれる
.github/skills/test-design/SKILL.md	ケース設計の手順。列定義は references/columns.md	Day3 演習6で作り、演習9で1行育てた	name とフォルダ名を同じ英小文字にする
.github/skills/regress/SKILL.md	貼った結果を前回と比べ、変わった行だけ返す	Day3 演習7	スキルはブラウザを開かない。結果を貼ってから呼ぶ
.github/agents/security-reviewer.agent.md	読み取り専用の見る役。tools を読み取り系だけにする	Day3 演習6。実技で新形式の部分を見させた	model はピッカーの表示を1文字も変えずに写す

公式コマンド（研修で使ったもの）

/create-skill 手順をスキルに切り出す。会話の流れからでも作れる

/create-instructions 特定のファイルだけに効く規約（applyTo）を作る

Chat: New Custom Agent コマンドパレットから、役割を1つに絞ったエージェントを作る

/clear 別の作業に移るとき。前の話を引きずらせない

/compact 会話が長くなって前提を忘れ始めたら、要点に畳んで続ける

/init フォルダの中身から、全体の指示ファイルの下書きを作る

/skills・/agents 入っているスキルとエージェントの一覧を確かめる

/troubleshoot スキルやエージェントが読まれないときに、ログを読ませる

Ask・Agent・Plan

ロール	ファイル	使いどころ
Ask	変わらない	中身を聞く、説明させる、案を比べる
Agent	Keep するまで確定しない	作る、直す、足す。違えばUndo
Plan	変わらない	大きな変更の前に段取りを確かめる

来月、形式の変わったCSVが届いたら

- 1 取り込む** index.html を開き、CSV を画面に落とす。警告欄の件数を見る
- 2 単体** tests.html を開き、SUMMARY 行を写す
SUMMARY PASS=n FAIL=n HOLD=n ...
- 3 通し** index.html?e2e で開き、E2E の結果を見る

- 4 並べる** SUMMARY 行と E2E の結果を貼ってから /regress /regress
 - 5 設計し直す** 形式が変わったら /test-design に仕様を添えてケースを出させ、worklog-core.js と worklog-config.js を直す
/test-design #新しい仕様.md
 - 6 見させる** 足した部分を security-reviewer に見させる
security-reviewer をサブエージェントで動かして、足した関数を見させて
- 呼び出しの型**
- 会話から作る** 通った流れをそのまま固定する
/create-skill ここまでの流れをスキルに。名前は regress
- 規約を作る** 効く範囲を先に決める
/create-instructions tests フォルダのファイルを扱うときだけ効く規約

並べて返させる 親の本文と区画を分けて返させる
security-reviewer をサブエージェントで動かし、結果を区画で返して

Tips・注意

Agent がテストのファイルを実装に合わせて書き換えてきたら、その差分は Keep しな。仕様と違う実装が PASS のまま残る。

期待値は testcases.md と仕様書から取り、worklog-core.js の中身から写さない。tests 用の規約に書いておけば言い忘れても効く。

1本の依頼文を保存する .prompt.md は移行先が SKILL.md と告知されている。依頼文1本でもスキルで作る。

サブエージェントは親より上のモデルでは動かない。親を上にして、見る役を軽くする。

外から取り込むスキルは、入れる前に SKILL.md と scripts を全部読む。scripts 付きは Python や Node が必要が多い。

Sources（公式・一次情報 / クリックで開く）

1. [VS Code 組み込みコマンド](#)
2. [VS Code Custom instructions](#)
3. [VS Code Agent Skills](#)
4. [VS Code Custom agents](#)
5. [VS Code Tools reference](#)
6. [vscode-copilot-chat /create-*](#)
- Copyright © 2026 Givery, Inc. All Rights Reserved.