

Harness Cheat Sheet of items commonly used at mechanical and electrical sites

Items used in maintenance, quality and design work, from 4 sources on 1 page / As of 2026-09



The 4 sources and how to choose

Source	Typical example	Prerequisites	Where to use it on site
Official commands	VS Code slash commands	Ready to use out of the box	Planning, and creating and managing the harness. Start here
Official templates	github/awesome-copilot (MIT, 423 skills and more)	If it is only SKILL.md, use it as it is. With scripts, Python or Node	Meeting minutes, reports and specifications. Copy them and adapt them for your own department
OSS	anthropics/skills (xlsx, pdf, docx, skill-creator and more)	Mostly Python. xlsx and others follow Anthropic's terms	A model for how to write skills. VS Code also reads .claude/skills/
Your own	The ones marked ★ that you built in the course	Extract from a conversation with /create-skill	Work your department repeats every month. It is the most effective

Official built-in commands (most used first)

/plan Have it produce only a plan before it starts working. Check the steps for totals and reports first

/compact When the conversation gets long, condense it to the key points and continue

/clear When you move to other work. Cut off the conversation about the previous machine

/init Creates or fixes the instructions file for the whole workspace from the contents of the folder

/create-prompt Lock in a request that worked as a command. Prompt files are moving to skills

/create-skill Bundle the steps, rules and templates into one folder. You can also create one from a conversation

/create-instructions Create rules that apply only to specific folders (applyTo)

/create-agent Create an agent limited to a single role, such as a read-only checker

/create-hook Register a process that must always run, such as a check before saving

/skills, /agents, /hooks Check the list of installed harness items and where they are stored

/models Open the model picker. Switch to a lighter model to save AI credits

/troubleshoot Have the AI read the debug log when a skill or agent is not being loaded

Harness location (workspace)

Location	What to write	Mechanical and Electrical track examples	Caution
AGENTS.md (top level)	Rules to follow in every conversation	Do not fill empty cells, do not open xlsx directly, do not write estimated numbers	No frontmatter needed. Keep it short
.github/instructions/*.instructions.md	Rules that apply only to the folders set by applyTo	applyTo: data/records/** with rules on how to write citations, dates and abbreviations	With neither applyTo nor description, it applies only when you attach it by hand
.github/skills/<name>/SKILL.md	Steps. Put rules in references/ and templates in templates/	operations-cleanup (data preparation), equipment-stop-report (report)	Make name the same as the folder name. Write "when to use it" in description
.github/agents/*.agent.md	An agent with a narrow role. Limit the tools it can use with tools	drawing-diff (drawing revision diff), doc-crosscheck (bill of materials and procurement request)	Limit the checker to read/ and search/ tools
.github/hooks/*.json	Processes that always run at a set point	Check for confidential terms before saving, keep a log of conversations	Needs shell execution. Check first whether company PCs allow it
~/copilot/skills/ and others	Your own skills (shared across all projects)	Your own meeting minutes template	Put anything the team uses on the workspace side

Skills worth building yourself (★ = created in the course)

clean-operations ★ When the Operations data for the month arrives. Fix units, full-width characters and inconsistent spellings. Do not fill empty cells

equipment-stop-report ★ Before the maintenance meeting. Report the one machine to stop, chosen by the decision rule, using the template

records-to-table ★ When inspection records and handover memos pile up. Merge them into one table that always has a source column

drawing-diff ★ (agent) When a drawing revision goes up. Differences in dimensions, fits, screws and keyways

doc-crosscheck ★ (agent) Before ordering. Mismatches in drawing number, revision, quantity per unit and quantity

ncr-report When a nonconformance occurs. Fields for symptom, cause, countermeasure and horizontal deployment

why-why When working out how to prevent recurrence. Ask why 5 times, with evidence at each level

fmea-draft When changing a design or process. A draft of the failure mode table. A person assigns the scores

work-instruction When handing work to someone. A standard work document with steps, key points and photo boxes

If you pick from official templates (awesome-copilot)

meeting-minutes Minutes of the maintenance meeting. Decisions, assignees and deadlines

incident-postmortem Review of an equipment problem (blameless format)

create-specification Specifications for improvement proposals and jigs

convert-excel-to-md, plus pdf and word versions Make xlsx, PDF and docx files readable (needs Python)

md-to-docx Submit reports as Word files (needs Node)

secrets-scanner, session-logger (hooks) Check for confidential terms and record conversations. Needs a shell

OSS anthropics/skills

xlsx Steps for reading and writing spreadsheets, formulas and charts. Assumes Python. Follow the license terms (no free redistribution)

skills/xlsx/SKILL.md

pdf, docx, pptx Extract from and fill in PDFs, and generate Word and PowerPoint files. The same license terms apply

skills/pdf/forms.md

skill-creator A skill for creating and evaluating skills. Apache 2.0

skills/skill-creator

Call patterns

Create from a conversation Lock in the flow that worked as it is

/create-skill Turn the flow so far into a skill. Name it records-to-table

Subagent Always include the name and where it should read
Run drawing-diff as a subagent and have it compare the 2 files in data/drawings/_svg

Chain them Have the skill call the checker at the end
After /operations-cleanup, have cleanup-checker check the CSV

Tips and cautions

Before you add a skill from outside, read all of SKILL.md and scripts, and check for instructions or commands you do not recognize.

On company PCs without administrator rights, skills with scripts often do not run. Try skills that have only SKILL.md first.

Sources (official and primary information / click to open)

1. [VS Code built-in commands](#)

2. [VS Code Agent Skills](#)

3. [VS Code Custom agents](#)

4. [VS Code Custom instructions](#)

5. [VS Code Hooks](#)

6. [github/awesome-copilot](#)

7. [anthropics/skills](#)

Copyright © 2026 Giverty, Inc. All Rights Reserved.