Three Claude Code skills worth installing, and a prompt for each

/statusline is already in Claude Code. /skill-creator and /claude-md-improver are Anthropic plugins, one install line each. This guide has the prompt Jason runs with each one, pointed at your own session history or your own CLAUDE.md.

GraniteAi, Manchester, New Hampshire. Facts checked 13 November 2026 against Anthropic's Claude Code docs and the claude-plugins-official repository on GitHub.
The guide page: graniteai.co/tools/claude-code-top-three-skills

START HERE

What you will have at the end

You watched the reel and you already use Claude Code. The three skills run in the reel's countdown order: number three first, number one last. Two of the six steps are install lines; the other four carry Jason's prompts.

- [ ] /skill-creator and /claude-md-improver installed from Anthropic's official plugin marketplace
- [ ] Jason's /statusline prompt run on your own sessions, with any status line script it writes saved in ~/.claude/
- [ ] Skill ideas drawn from one project's session history, from /skill-creator
- [ ] A quality report on a project CLAUDE.md, with every proposed change shown as a diff for your OK
- [ ] The same audit on your global ~/.claude/CLAUDE.md

What is built in and what you install

Skill	Where it comes from	What you do first
/statusline	Built into Claude Code.	Nothing. Send the prompt.
/skill-creator	The skill-creator plugin, in Anthropic's official marketplace.	One install line.
/claude-md-improver	A skill in Anthropic's claude-md-management plugin, same marketplace.	One install line.

Anthropic makes all three. GraniteAi made none of them; the prompts are Jason's. Claude Code adds Anthropic's official marketplace for you the first time you start an interactive terminal session.

NUMBER THREE

/statusline

It is built into Claude Code, so there is nothing to install. You describe what you want in plain words, and Claude Code writes a script file in **~/.claude/** and updates your settings. The status line shows in its own row above the built-in footer badges, which stay where they are.

Send in Claude Code

```
/statusline identify the 3 highest impact UX changes we can make based on session transcripts and my model correction and steering patterns
```

Your part: read what it proposes before you approve a change. This prompt asks for more than a status bar: it has Claude Code look at how you correct and steer the model and propose status line changes from that. In the reel, Claude Code gave it to a background agent that finished in 2 minutes 11 seconds. What comes back depends on your sessions.

Docs: code.claude.com/docs/en/statusline

NUMBER TWO

/skill-creator

It comes from the skill-creator plugin in Anthropic's official marketplace. One line installs it.

Send in Claude Code

```
/plugin install skill-creator@claude-plugins-official
```

Your part: send the line and approve anything Claude Code asks. In VS Code, if the plugin is not found, add anthropics/claude-plugins-official on the Marketplaces tab first.

Then run it from inside a project. Jason uses it for any workflow that comes down to the same series of actions, repeated over and over. His prompt has it read this project's session transcripts on disk and name the patterns where a skill would save time, improve accuracy or improve outcomes against the project's goal.

Send in Claude Code

```
/skill-creator search the session transcripts on disk for this project's history and identify the patterns which emerge that demonstrate a specific skill will save the user time, improve accuracy of work, or enhance outcomes relative to the project's goal
```

Your part: open Claude Code in the project you want it to study, then decide which of the skills it suggests are worth building. Transcripts are kept 30 days by default, so it reads about the last month.

Source: skill-creator on GitHub, by Anthropic

NUMBER ONE

The CLAUDE.md improver

/claude-md-improver is a skill in Anthropic's claude-md-management plugin. It scans for CLAUDE.md files, evaluates them against templates, writes a quality report, then makes targeted updates. It asks for confirmation before it updates anything.

Send in Claude Code

```
/plugin install claude-md-management@claude-plugins-official
```

Your part: send the line and approve anything Claude Code asks. The plugin is named claude-md-management; the skill inside it is claude-md-improver.

Run it in a project first

Project level means **./CLAUDE.md** or **./.claude/CLAUDE.md** in that repo. The parts in angle brackets are yours to fill in before you send it: the file to audit, who you are, your goals in order, and what done means for your work.

Fill in the angle brackets, then send in Claude Code

```
/claude-md-improver Audit <path to CLAUDE.md, or "this repo">.

Context the file should serve:
- Who I am: <role, company legal name, clients>
- My goals, in priority order: <1) ... 2) ... 3) ...>
- What "done" means for my work: <e.g. seen running, rendered, deployed>

How to audit:
1. Check every path, command, skill, plugin, hook and tool the file names against the
   real system (filesystem, settings.json, enabled plugins, PATH). Flag anything dead.
2. Find rules that sit at the wrong level: global rules that only fit some repos, and
   repo rules repeated globally.
3. Check memory files and files the CLAUDE.md links to (commands, sibling CLAUDE.md
   files) for facts that contradict it.
4. Tell me what the file is missing for my goals above, not only what's wrong with it.

Rules for changes:
- Back up the file before editing.
- Show every change as a diff and wait for my OK.
- You may fix files that directly depend on a change (commands, memories). List each one.
- Don't commit. Leave third-party and plugin-managed files alone; report their issues.
- Keep the file as short as or shorter than it is now, unless a line adds what my goals need.
```

Your part: replace every <...> with your own details, and leave none in. The prompt tells it to back up the file, show each change as a diff, wait for your OK and not commit.

NUMBER ONE, SECOND RUN

Then run it at the global level

That is **~/.claude/CLAUDE.md**, the file in your root Claude Code folder. On Windows it is %USERPROFILE%\.claude\CLAUDE.md. Claude Code reads it in every project, together with that project's own CLAUDE.md: the files are added to each other, and neither one overrides the other.

Fill in the angle brackets, then send in Claude Code

```
/claude-md-improver Audit ~/.claude/CLAUDE.md.

Context the file should serve:
- Who I am: <role, company legal name, clients>
- My goals, in priority order: <1) ... 2) ... 3) ...>
- What "done" means for my work: <e.g. seen running, rendered, deployed>

How to audit:
1. Check every path, command, skill, plugin, hook and tool the file names against the
   real system (filesystem, settings.json, enabled plugins, PATH). Flag anything dead.
2. Find rules that sit at the wrong level: global rules that only fit some repos, and
   repo rules repeated globally.
3. Check memory files and files the CLAUDE.md links to (commands, sibling CLAUDE.md
   files) for facts that contradict it.
4. Tell me what the file is missing for my goals above, not only what's wrong with it.

Rules for changes:
- Back up the file before editing.
- Show every change as a diff and wait for my OK.
- You may fix files that directly depend on a change (commands, memories). List each one.
- Don't commit. Leave third-party and plugin-managed files alone; report their issues.
- Keep the file as short as or shorter than it is now, unless a line adds what my goals need.
```

Your part: fill in the same placeholders. Point 2 of the audit matters most at this level: it looks for global rules that only fit some repos, and repo rules repeated globally.

Docs: CLAUDE.md levels, in the Claude Code docs. Source: claude-md-management on GitHub, by Anthropic

When something goes wrong

What you see	What it means	The fix
/skill-creator or /claude-md-improver does nothing	The plugin did not install, or another command already uses the short name.	Check the plugin in /plugin, then use the full name: /skill-creator:skill-creator or /claude-md-management:claude-md-improver.
The install line says the plugin is not found	The official marketplace is not added yet. That can happen in VS Code.	Add anthropics/claude-plugins-official on the Marketplaces tab, then send the install line again.
The transcript prompts find little to work with	Transcripts are kept 30 days by default, and each project has its own.	Run the prompt from a project you have worked in during the last month.
The CLAUDE.md audit asks who you are	An angle-bracket placeholder was left in the prompt.	Fill in every <...> and send it again.
You want a change undone	The prompt has Claude Code back up the file before editing and wait for your OK on each diff.	Restore the backup it made, or say no to the diff.

The specifics

Question	Answer
Where the status line lives	/statusline writes a script file in ~/.claude/ and updates your settings. It shows in its own row above the footer badges.
Where CLAUDE.md lives	Project: ./CLAUDE.md or ./.claude/CLAUDE.md. Global: ~/.claude/CLAUDE.md. Claude Code loads both, one added to the other.
Session history	Transcripts are .jsonl files in ~/.claude/projects/, kept 30 days by default, so these prompts see about the last month.
VS Code	The steps follow the terminal. We have not tested them in VS Code.

KEEP GOING

The guide page

Every step, prompt and install line in this PDF lives on the page too, with the latest checked facts: graniteai.co/tools/claude-code-top-three-skills

See what else GraniteAi makes: graniteai.co/tools/claude-code-top-three-skills#for-sale