

Claude Code Slash Commands Cheat Sheet

Your 12 must-know commands to go hands-off and build 2× faster in Claude Code.

01

1. `/loop` — Run on repeat, hands-off

Hand Claude one instruction like “run the tests and fix the failures” and it keeps taking passes until the job is done. The session stays open so you never re-prompt after every attempt.

02

2. `/background` — Keep working after you leave

The agent keeps building even after you close the session. Kick off a big refactor, shut the laptop, and come back to finished work — no sitting and watching a spinner.

03

3. `/context` — See the memory fill live

Shows your context window filling up in real time so you can spot the moment Claude is about to start forgetting earlier parts of the conversation — and act before quality slips.

QUICK TIPS

- Watch for the red zone, then summarize or clear before you push more work in.

04

4. `/usage` — Know exactly where you stand

Your dashboard for cost, plan limits, and stats in one place. Stop guessing how much of your allowance you’ve burned through this session.

05

5. `/branch` — Test two paths, lose neither

Splits your conversation into two approaches so you can try Approach A and Approach B in parallel, then keep whichever one wins. Perfect when you want to experiment without losing a good thread.

06

6. `/model` — Switch AI mid-task

Run a stronger model like Opus for the genuinely hard problems, then flip to a faster one like Haiku for simple, repetitive work — without leaving your task. Speed where it doesn't matter, horsepower where it does.

07

7. `/security-review` — Catch holes before they ship

Scans your files for security problems — exposed secrets, shaky auth, risky config — and flags them so you fix issues before they go live, not after.

QUICK TIPS

- Run it right before every push. Non-negotiable if you're shipping real apps.
-

08

8. `/init` — Teach Claude your project

Scaffolds a `CLAUDE.md` file that documents your codebase, conventions, and rules. Once it exists, Claude understands your project on every future session instead of starting blind.

09

9. `/clear` — Fresh start between tasks

Wipes the conversation context so an old task doesn't bleed into a new one. Use it whenever you switch to something unrelated and want Claude focused and fast.

QUICK TIPS

- A clean context beats a bloated one — clear early, clear often.
-

10

10. `/agents` — Bring in specialized help

Spins up focused subagents for specific jobs — research, review, planning — so heavy tasks run in their own clean context instead of clogging your main thread.

11

11. `/rewind` — Go back to before it broke

Restores your code and conversation to an earlier checkpoint. When an edit sends things sideways, jump back to the last good version instead of untangling the mess by hand.

12. /review — Get a full code review

Runs a structured review of your changes and surfaces bugs, simplifications, and cleanups — like having a senior engineer read your diff before you merge.

Ready to go deeper?

This cheat sheet is just the start. Inside Vibe Coding Mastery you get the full agentic Claude Code workflows, master prompts, and step-by-step build walkthroughs so you can ship real systems — not just run commands.

[Join the Vibe Coding Build →](https://builds.digicuratoragency.com/)

<https://builds.digicuratoragency.com/>