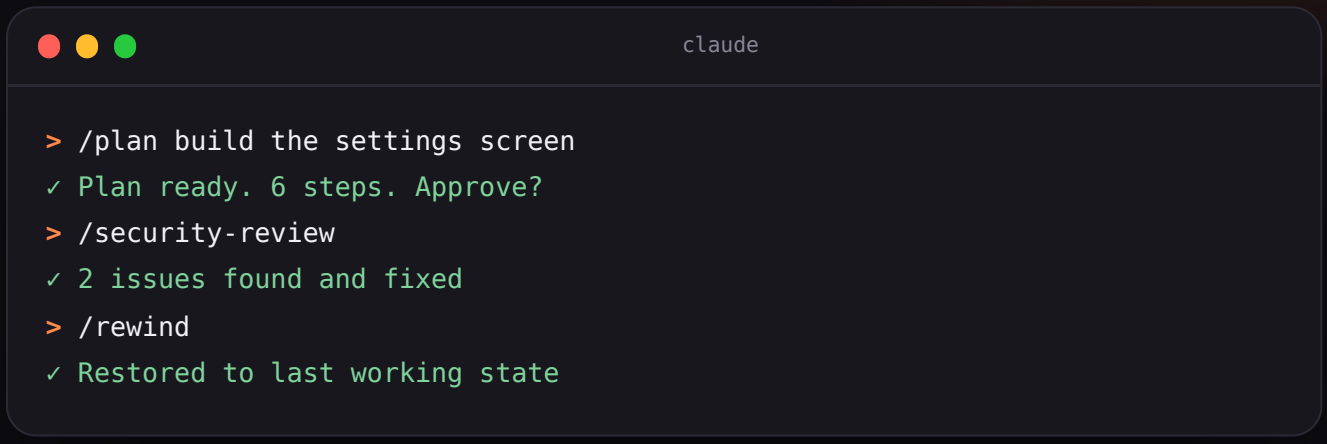
FREE GUIDE · FOR DEVELOPERS & BUILDERS

24 Claude Code tricks *most people* never use

Stop using Claude Code like a beginner. Every command here is real, copy-paste ready, and explained in one minute.



24
tricks with examples

5
chapters

1
page cheat sheet

Somya Modi
Founder, Modi Studio · building Vayli with Claude Code

Instagram @s0my.modi
November 2026 edition

BEFORE YOU START

How to get the most from this guide

Claude Code is an AI that works inside your project folder. It reads your files, writes code, runs commands and fixes errors. Most people only type requests into it. The commands below are what make it feel like a full team.

Open it

Open a terminal in your project folder and type `claude`. Everything in this guide is typed inside that session.

See every command

Type `/` to see the full list. Type `/` plus a few letters to filter it.

Start small

Learn one chapter a day. Try each example on a real project, not a test file.

Stay updated

Claude Code updates often. If a name changes, the `/` list and `/release-notes` always show the latest.

The 5 from the reel

#	TRICK	COMMAND	PAGE
01	Parallel Claude	`git worktree`	3
16	Connect tools with MCP	`/mcp`	6
20	Claude as your teacher	`/output-style`	6
12	Find security holes	`/security-review`	5
11	Undo Claude's mistakes	`/rewind`	5

Tip: cards marked IN THE REEL are the ones I showed in the video. Start with those.

CHAPTER 01

01 Work faster

Do more in the same hour.

01 Parallel Claude with Git worktrees

IN THE REEL `git worktree`

Run three Claude sessions on one project at the same time. Each one works in its own copy of the code, so they never overwrite each other.

Use it when: One Claude fixes a bug, one builds a feature, one writes tests.

```
git worktree add ../myapp-fix -b fix-login
cd ../myapp-fix && claude
```

02 Plan before you code

`Shift + Tab · /plan`

Plan mode makes Claude read the code and show you a step-by-step plan first. Nothing changes until you approve it.

Use it when: Big changes, new features, anything you don't want broken.

```
/plan add Google login to the signup screen
```

03 Ask a side question

`/btw`

Ask a quick question without adding it to the main chat history, so your current task stays clean.

Use it when: You're mid-task and just need a fact or an explanation.

```
/btw what does useEffect cleanup do?
```

04 Keep going until it's done

`/goal`

Give Claude a finish line. It keeps working across turns until the condition is met.

Use it when: Fixing all failing tests, clearing every lint error.

```
/goal every test in tests/ passes
```

05 Repeat a task on a timer

`/loop`

Runs the same prompt again and again at an interval while the session is open.

Use it when: Watching a build, checking logs, polling for errors.

```
/loop 5m check the build and report any errors
```

PRO TIP

Combine them: run /plan first, then let a second Claude build the plan in its own worktree while you keep working.

CHAPTER 02

02 Memory & context

Stop explaining the same thing twice.

06 Give Claude a project guide

`/init · CLAUDE.md`

/init reads your project and writes a CLAUDE.md file: what the app does, how to run it, your rules. Claude reads it in every new chat.

Use it when: The first time you open any project.

```
/init
/memory   # edit CLAUDE.md later
```

07 Shrink a long chat

`/compact`

Summarises the conversation so Claude stops forgetting things and you save usage. You can tell it what to keep.

Use it when: Long sessions where answers start getting worse.

```
/compact keep the API decisions and the
open bugs
```

08 See what Claude remembers

`/context`

Shows a coloured grid of what is filling Claude's context window and how much space is left.

Use it when: Chat feels slow or Claude starts missing details.

```
/context
```

09 Pick up where you left off

`/resume · claude -c`

Reopen yesterday's conversation with all its context. claude -c jumps straight into your last one.

Use it when: Every morning, or after closing the terminal.

```
claude -c
/resume   # pick from a list
```

10 Branch the conversation

`/branch · /fork`

Split the chat at this point and try another approach, without losing the original path.

Use it when: Comparing two designs or two fixes.

```
/branch try-redux
```

PRO TIP

Run /init again after big changes so CLAUDE.md stays up to date. A good CLAUDE.md is the biggest quality boost you can get for free.

CHAPTER 03

03 Safety & quality

Ship code you can trust.

11 Undo Claude's mistakes

IN THE REEL `/rewind · Esc Esc`

Rolls the conversation and/or your code back to an earlier checkpoint. Press Esc twice to open it fast.

Use it when: Claude broke something and you want the last working state.

```
/rewind
```

12 Find security holes

IN THE REEL `/security-review`

Checks your branch changes for security problems like leaked keys and unsafe input before they go live.

Use it when: Before every release or pull request.

```
/security-review
```

13 Get a code review

`/code-review · /review`

Reviews your current changes, a branch or a PR and points out bugs and risky code.

Use it when: Before you merge anything.

```
/code-review
```

14 See exactly what changed

`/diff`

Shows every change in your working tree, including Claude's edits, so nothing slips in unseen.

Use it when: After a long session, before you commit.

```
/diff
```

15 Clean up messy code

`/simplify`

Finds cleanup chances in the code you changed and applies the fixes.

Use it when: Code works but looks long or repeated.

```
/simplify src/components
```

PRO TIP

Commit often. /rewind is great for Claude's edits, but Git is still your real safety net.

CHAPTER 04

04 Superpowers

Turn Claude into a team.

16 Connect your tools with MCP

IN THE REEL `/mcp · claude mcp add`

MCP plugs Claude into other tools: browsers, databases, GitHub and more. Claude then works inside them directly.

Use it when: You want Claude to test your site, read your DB or open issues.

```
claude mcp add playwright -- npx
@playwright/mcp@latest
/mcp   # manage connections
```

17 Build your own AI team

`/agents`

Subagents are specialists with their own instructions: a bug hunter, a UI reviewer, a test writer.

Use it when: Repeated jobs that need a fixed expert mindset.

```
/agents   # create or manage subagents
```

18 Teach Claude your workflow

`Skills`

Save a process once as a skill (.claude/skills/<name>/SKILL.md). Claude follows it every time, and you can call it with /name.

Use it when: Deploy steps, release notes, your coding style.

```
.claude/skills/release/SKILL.md
/skills   # list your skills
```

19 Automate with hooks

`/hooks`

Run a command automatically at set moments, like formatting the code every time Claude edits a file.

Use it when: Formatting, linting or tests you always forget.

```
"hooks": { "PostToolUse": [{ "matcher":
"Edit|Write",
  "hooks": [{ "type": "command",
"command": "npx prettier --write ." }]
}] }
```

20 Make Claude your teacher

IN THE REEL `/output-style`

Switch to the Learning style and Claude explains each step while it codes, sometimes asking you to write small parts.

Use it when: You're learning to code, not just shipping.

```
/output-style learning
```

PRO TIP

Switch off MCP servers you are not using in /mcp. Fewer tools means a cleaner context and faster answers.

CHAPTER 05

05 Pro shortcuts

Small habits, big speed.

21 Use Claude from scripts

`claude -p`

Print mode: Claude answers one prompt and exits. You can pipe files and logs straight into it.

Use it when: Quick fixes, scripts, CI jobs.

```
cat error.log | claude -p "explain this
error and the fix"
```

22 Point, run and paste

`@ · ! · Ctrl + V`

@ adds a file to the prompt, ! runs a terminal command right in the chat, and you can paste a screenshot so Claude sees the bug.

Use it when: Every single day.

```
@src/app.tsx why is this slow?
! npm test
```

23 Control speed and cost

`/effort · /usage`

/effort sets how hard Claude thinks. /usage shows your session cost and plan limits.

Use it when: Simple task: lower effort. Hard bug: raise it.

```
/effort low
/usage
```

24 Stop the permission pop-ups

`/fewer-permission-prompts`

Adds an allowlist for the safe, read-only actions you keep approving, so Claude asks less often.

Use it when: When you keep clicking "yes" for the same thing.

```
/fewer-permission-prompts
```

PRO TIP

Type / and a few letters to find any command in a second. You never have to memorise them.

PUT IT TOGETHER

My daily Claude Code workflow

1	`claude -c`	Continue yesterday's session with full context.
2	`/plan`	Get a plan for today's feature and approve it.
3	`git worktree`	Run a second Claude on bugs while the first builds.
4	`/diff + /code-review`	Check every change before it goes in.
5	`/security-review`	Last check before release. Then ship.

CHEAT SHEET

All 24 tricks on one page

Print it, keep it next to your keyboard.

#	COMMAND	WHAT IT'S FOR
01	`git worktree`	Parallel Claude with Git worktrees
02	`Shift + Tab · /plan`	Plan before you code
03	`/btw`	Ask a side question
04	`/goal`	Keep going until it's done
05	`/loop`	Repeat a task on a timer
06	`/init · CLAUDE.md`	Give Claude a project guide
07	`/compact`	Shrink a long chat
08	`/context`	See what Claude remembers
09	`/resume · claude -c`	Pick up where you left off
10	`/branch · /fork`	Branch the conversation
11	`/rewind · Esc Esc`	Undo Claude's mistakes
12	`/security-review`	Find security holes
13	`/code-review · /review`	Get a code review
14	`/diff`	See exactly what changed
15	`/simplify`	Clean up messy code
16	`/mcp · claude mcp add`	Connect your tools with MCP
17	`/agents`	Build your own AI team
18	`Skills`	Teach Claude your workflow
19	`/hooks`	Automate with hooks
20	`/output-style`	Make Claude your teacher
21	`claude -p`	Use Claude from scripts
22	`@ · ! · Ctrl + V`	Point, run and paste
23	`/effort · /usage`	Control speed and cost
24	`/fewer-permission-prompts`	Stop the permission pop-ups

ONE LAST THING

I build my own apps with *exactly these* tricks.

Vayli, the notch for Windows, is built with Claude Code using this workflow. It's live on the Microsoft Store with a 3-day free trial.

Follow for more
Instagram **@s0my.modi**

Try Vayli
Search **Vayli** on the Microsoft Store

Found this useful? Share it with one friend who still uses Claude Code like a beginner.

Somya Modi · Founder, Modi Studio

Commands checked against the Claude Code commands reference in November 2026. Claude Code changes often; type / in Claude Code for the current list. Not affiliated with Anthropic.