



// QUICK START

Type `/init` and add your project rules. Type `/memory` and add your preferences. Type `/compact` when sessions get long. That's the foundation. Everything else builds on those three.

01 BUILD PERSISTENT SYSTEMS

COMMAND	WHAT IT DOES	WHEN TO USE
<code>/init</code>	Creates project memory (CLAUDE.md) that Claude reads every session	Starting a new project
<code>/memory</code>	Stores personal preferences that follow you across all projects	Once, then update as needed
<code>/permissions</code>	Controls which files and commands Claude can access	Setting up client or production work
<code>/compact</code>	Compresses conversation context, keeps key decisions	Sessions feel sluggish
<code>/context</code>	Visual grid showing what's using your context window	Before compacting
<code>/resume</code>	Pick up a previous session by name with full context	Returning to ongoing work

02 PLAN & SHIP

COMMAND	WHAT IT DOES	WHEN TO USE
<code>/plan</code>	Structured planning before execution. Claude reads, plans, then asks for approval	Changes touch 2+ files
<code>/ultraplan</code>	Architecture-level planning with browser review before execution	Large features or refactors
<code>/rewind</code>	Checkpoint and rollback. Code changes revert too	Wrong approach, need to undo
<code>/diff</code>	Interactive viewer showing every change Claude made	Before committing
<code>/batch</code>	Parallel work across your codebase. Each unit gets its own PR	Large-scale changes
<code>/autofix-pr</code>	Watches your PR and auto-fixes CI failures and review comments	Code teams with PR workflows

03 WORK FROM ANYWHERE

COMMAND	WHAT IT DOES	WHEN TO USE
<code>/desktop</code>	Move terminal session to the desktop app	Want the desktop UI
<code>/teleport</code>	Pull a web session into your local terminal	Need local filesystem access
<code>/remote-control</code>	Control your session from claude.ai on any device	Working away from your desk
<code>/voice</code>	Push-to-talk voice dictation. Talk instead of type	Thinking out loud

SKIP THESE (FOR NOW)

`/mcp /agents /hooks /sandbox /setup-bedrock /setup-vertex ...` power user territory. You'll know when you need them. Also nice to know: `/export /powerup /insights /stats /cost /usage`



04

CUSTOM SKILLS

Custom skills are markdown files in `.claude/skills/` that encode your workflows into repeatable commands. A coach who onboards every client the same way. A consultant who generates the same report. An agency owner who audits the same checklist. Type `/your-skill-name` and Claude executes it every time.

```
# .claude/skills/client-report/SKILL.md

---
name: client-report
description: Generate a weekly client performance report
allowed-tools: Read, Grep, Bash(git:*)
---
```

Generate a weekly performance report for the client. Include:

1. Ad spend vs. budget (pull from the analytics export)
2. Cost per lead this week vs. last week
3. Top 3 performing creatives with spend and CPA
4. One specific recommendation based on the data

Format as markdown. Keep it under 500 words.
Use exact numbers, not percentages alone.

SCOPE	LOCATION	WHO USES IT
Personal	<code>~/.claude/skills/your-skill/SKILL.md</code>	You, across all projects
Project	<code>.claude/skills/your-skill/SKILL.md</code>	Anyone on this project
Plugin	Installed via plugin system	Anyone who installs the plugin

05

KEY FRONTMATTER FIELDS

<p>allowed-tools</p> <p>Restricts which tools the skill can access. Limits blast radius.</p>	<p>model</p> <p>Which AI model to use. Faster models for simple tasks save time.</p>	<p>disable-model-invocation</p> <p>Set to true so the skill only runs when you type it.</p>
---	---	---

THE RESET TAX

5-10 min/session × 3-4 sessions/day = **100-150 hours/year**

At \$100-\$200/hr = **\$10,000-\$30,000/year** on context recovery

Slash commands eliminate this. `/init` + `/memory` + `/resume` = sessions that compound instead of reset.

06

PRO TIPS

<p>TREAT CLAUDE.MD AS LIVING</p> <p>Every time you re-explain something, add it to the file. It compounds over weeks.</p>	<p>COMPACT WITH FOCUS</p> <p><code>/compact keep the auth decisions</code> tells Claude what to prioritize when compressing.</p>
<p>NAME YOUR SESSIONS</p> <p><code>/rename client-dashboard</code> then <code>/resume</code> it days later with full context.</p>	<p>PLAN BEFORE BUILDING</p> <p>Use <code>/plan</code> on anything that touches more than two files. Fewer wrong turns.</p>