

Claude Code Cheatsheet

Daily essentials for maximum productivity

Florian BRUNIAUX

February 2026

v3.27.5



Table of contents

1	How to Install	2
2	Essential Commands	2
3	Keyboard Shortcuts	3
4	File References	4
5	Overlooked Features (But Official!)	4
6	Permission Modes	5
7	Memory & Settings (2 levels)	5
7.1	.claude/ Folder Structure	6
8	Context Management (CRITICAL)	6
8.1	Context Recovery Commands	8
9	Plan Mode & Thinking	8
10	Typical Workflow	9
11	Quick Prompting Formula	9
12	MCP Servers	10
13	CLI Flags Quick Reference	10
14	Anti-patterns	11
15	Cost Optimization	12
16	Quick Decision Tree	12
17	Common Issues Quick Fix	13

1 How to Install

```
npm install -g @anthropic-ai/claude-code  
# Verify installation  
which claude && claude --version
```

Health Check: `claude doctor && claude mcp list`

2 Essential Commands

Command	Action
<code>/help</code>	Contextual help
<code>/clear</code>	Reset conversation
<code>/compact</code>	Free up context
<code>/status</code>	Session state + context usage
<code>/context</code>	Detailed token breakdown
<code>/plan</code>	Enter Plan Mode (no changes)
<code>/execute</code>	Exit Plan Mode (apply changes)
<code>/model</code>	Switch model (sonnet/opus/opusplan)
<code>/insights</code>	Usage analytics + optimization
<code>/teleport</code>	Teleport session from web
<code>/tasks</code>	Monitor background tasks

Command	Action
<code>/fast</code>	Toggle fast mode (2.5x speed, 6x cost)
<code>/debug</code>	Systematic troubleshooting
<code>/remote-env</code>	Configure cloud environment
<code>/exit</code>	Quit (or Ctrl+D)

3 Keyboard Shortcuts

Shortcut	Action
<code>Shift+Tab</code>	Cycle permission modes
<code>Esc × 2</code>	Rewind (undo)
<code>Ctrl+C</code>	Interrupt
<code>Ctrl+R</code>	Search command history
<code>Ctrl+L</code>	Clear screen (keeps context)
<code>Ctrl+B</code>	Background tasks
<code>Alt+T</code>	Toggle thinking on/off
<code>Tab</code>	Autocomplete
<code>Shift+Enter</code>	New line
<code>Ctrl+D</code>	Exit

IDE Shortcuts: VS Code `Alt+K` | JetBrains `Cmd+Option+K`

4 File References

```
@path/to/file.ts    → Reference a file
@agent-name         → Call an agent
!shell-command      → Run shell command
```

5 Overlooked Features (But Official!)

Feature	Since	What It Does
Tasks API	v2.1.16	Persistent task lists with dependencies
Background Agents	v2.0.60	Sub-agents work while you code
Agent Teams	v2.1.32	Multi-agent coordination (TeamCreate/SendMessage)
Auto-Memories	v2.1.32	Automatic cross-session context capture
Session Forking	v2.1.19	Rewind + create parallel timeline
LSP Tool	v2.0.74	Code intelligence (go-to-def, refs)
Rules Templates	v3.27.4	4 ready-to-use <code>.claude/rules/</code> templates (arch/quality/test/perf)
/review-plan	v3.27.4	Structured plan review across 4 axes before coding

Pro tip: These aren't "secrets" — they're in the [CHANGELOG](#). Read it!

6 Permission Modes

Mode	Editing	Execution
Default	Asks	Asks
Auto-accept	Auto	Asks
Plan Mode	None	None

Shift+Tab to switch modes

7 Memory & Settings (2 levels)

Level	Path	Scope	Git
Project	<code>.claude/</code>	Team	Yes
Personal	<code>~/.claude/</code>	You (all projects)	No

Priority: Project overrides Personal

7.1 .claude/ Folder Structure

```
.claude/
├── CLAUDE.md          # Local memory (gitignored)
├── settings.json       # Hooks (committed)
├── settings.local.json # Permissions (not committed)
├── agents/            # Custom agents
├── commands/          # Slash commands
├── hooks/             # Event scripts
├── rules/             # Auto-loaded rules (v3.26+)
│   ├── architecture-review.md # ← templates ready in examples/rules/
│   ├── code-quality-review.md
│   ├── test-review.md
│   └── performance-review.md
└── skills/            # Knowledge modules
```

8 Context Management (CRITICAL)

Statusline: Model: Sonnet | Ctx: 89.5k | Ctx(u): 56.0%

✅ 0-50%	⚠️ 50-70%	💎 70-90%	🏆 90%+

Watch `ctx(u):` → >70% = `/compact`, >85% = `/clear`

Sign	Action
Short responses	<code>/compact</code>
Frequent forgetting	<code>/clear</code>
>70% context	<code>/compact</code>
Task complete	<code>/clear</code>

8.1 Context Recovery Commands

Command	Usage
<code>/compact</code>	Summarize and free context
<code>/clear</code>	Fresh start
<code>/rewind</code>	Undo recent changes
<code>claude -c</code>	Resume last session
<code>claude -r <id></code>	Resume specific session

9 Plan Mode & Thinking

Feature	Activation	Usage
Plan Mode	<code>Shift+Tab × 2</code> or <code>/plan</code>	Explore without modifying
OpusPlan	<code>/model opusplan</code>	Opus for planning, Sonnet for execution

Opus 4.6: Thinking is ON by default at max budget. Keywords like “ultrathink” are cosmetic only.

Control	Action	Persistence
Alt+T	Toggle thinking on/off	Session
/config	Enable/disable globally	Permanent
effort param	API only: low/medium/high/max	Per-request

Cost tip: For simple tasks, Alt+T to disable thinking → faster & cheaper.

10 Typical Workflow

1. Start session → `claude`
2. Check context → `/status`
3. Plan Mode → `Shift+Tab × 2` (for complex tasks)
4. Describe task → Clear, specific prompt
5. Review changes → Always read the diff!
6. Accept/Reject → `y/n`
7. Verify → Run tests
8. Commit → When task complete
9. `/compact` → When context >70%

11 Quick Prompting Formula

WHAT: [Concrete deliverable]
WHERE: [File paths]
HOW: [Constraints, approach]
VERIFY: [Success criteria]

Example:

```
Add input validation to the login form.  
WHERE: src/components/LoginForm.tsx  
HOW: Use Zod schema, show inline errors  
VERIFY: Empty email shows error, invalid format shows error
```

12 MCP Servers

Server	Purpose
Serena	Indexing + session memory + symbol search
grepai	Semantic search + call graph analysis
Context7	Library documentation
Sequential	Structured reasoning
Playwright	Browser automation
Postgres	Database queries
doobidoo	Semantic memory + Knowledge Graph

Serena memory: `write_memory()` / `read_memory()` / `list_memories()`

Check status: `/mcp`

13 CLI Flags Quick Reference

Flag	Usage
<code>-p "query"</code>	Non-interactive mode (CI/CD)

Flag	Usage
<code>-c</code> / <code>--continue</code>	Continue last session
<code>-r</code> / <code>--resume <id></code>	Resume specific session
<code>--teleport</code>	Teleport session from web
<code>--model sonnet</code>	Change model
<code>--add-dir ../lib</code>	Allow access outside CWD
<code>--permission-mode plan</code>	Plan mode
<code>--dangerously-skip-permissions</code>	Auto-accept (use carefully)
<code>--mcp-debug</code>	Debug MCP servers
<code>--allowedTools "Edit,Read"</code>	Whitelist tools
<code>--debug</code>	Debug output

14 Anti-patterns

❌ Don't

- Vague prompts
- Accept without reading
- Ignore warnings
- Skip permissions
- Negative constraints only

✅ Do

- Specify file + line with @references
- Read every diff
- Use /compact at 70%
- Never in production
- Provide alternatives

15 Cost Optimization

Model	Use For	Cost
Haiku	Simple fixes, reviews	\$
Sonnet	Most development	\$\$
Opus	Architecture, complex bugs	\$\$\$
OpusPlan	Plan (Opus) + Execute (Sonnet)	\$\$

16 Quick Decision Tree

Simple task	→ Just ask Claude
Complex task	→ Tasks API to plan first
Risky change	→ Plan Mode first
Repeating task	→ Create agent or command
Context full	→ /compact or /clear
Need docs	→ Use Context7 MCP
Deep analysis	→ Use Opus (thinking on by default)

The Golden Rules

- 1 Always review diffs before accepting
- 2 Use /compact before context gets critical (>70%)
- 3 Be specific in requests (WHAT, WHERE, HOW, VERIFY)
- 4 Plan Mode first for complex/risky tasks
- 5 Create CLAUDE.md for every project

6

Commit frequently after each completed task

7

Know what's sent — prompts, files, MCP results → Anthropic

17 Common Issues Quick Fix

Problem	Solution
"Command not found"	Check PATH, reinstall npm global
Context too high (>70%)	<code>/compact</code> immediately
Slow responses	<code>/compact</code> or <code>/clear</code>
MCP not working	<code>claude mcp list</code> , check config
Permission denied	Check <code>settings.local.json</code>
Hook blocking	Check hook exit code, review logic

Health Check: `which claude && claude doctor && claude mcp list`

Author: Florian BRUNIAUX | [Methode Aristote](#) | Written with Claude

Version 3.27.4 | February 2026