

Claude Code Cheatsheet

Quick Reference Guide for Maximum Productivity

Essential Keybindings

Shift + Tab	Auto-accept edits
#	Create a memory
!	Enter bash mode
@	Add file/folder to context
Esc	Cancel operation
Esc Esc	Jump back in history / Resume
Ctrl + R	Verbose output mode
/vibe	Check Claude's understanding
/clear	Clear conversation context

Configuration Files

Claude uses configuration files to know what to do and how to behave. The key files include `claude.md` and custom slash commands.

- claude.md** — Defines what Claude should know about your project.
- Custom Slash Commands** — Define what Claude should do, including tasks and workflows.

Subagent Benefits

Context Preservation	Each subagent operates independently, preventing cross-task interference.
Specialized Expertise	Fine-tuned for specific technical or domain tasks.
Reusability	Reusable across projects and teams.
Flexible Permissions	Custom tool access for each subagent type.

Prompt Structure (10 Components)

Craft effective prompts using these ten structured components for clarity, consistency, and precision.

- Task Context
- Tone Context
- Background Data
- Detailed Task
- Examples
- Conversation History
- Immediate Task
- Thinking Prompt
- Output Formatting
- Prefilled Response

Best Practices

Follow these do's and don'ts to ensure smooth collaboration and effective code assistance.

DO: Provide clear context, iterate often, use version control, and share visuals.

DON'T: Skip planning, mix unrelated tasks, or assume Claude knows implicit context.

Key Philosophy

Focus on solutions, not syntax. Master your tools, and let Claude handle complexity while you stay creative.

Quick Links

Docs	https://docs.anthropic.com
API	https://docs.anthropic.com/en/api
GitHub	https://github.com/anthropics/claude-code