Cheatography

Claude Code Cheat Sheet

by Dave Child (DaveChild) via cheatography.com/1/cs/48210/

What is Claude Code?

An agentic CLI tool that reads your codebase, edits files, runs shell commands, and manages git workflows. It works across your terminal, VS Code, JetBrains, a desktop app, and the web.

It is not a chat interface that suggests code.

Claude plans work, writes across multiple files, runs tests, and creates PRs. It can take actions you can't undo — permissions matter.

Installing Claude

```
# Recommended (auto-updates)
curl -fsSL https: //c lau de.a - i/ ins tall.sh | bash
# macOS Homebrew
brew install --cask claude -code
```

First Run

```
cd your-project
claude # intera ctive session
claude auth login # if not authen ticated
```

On first run Claude reads your repo and prompts for auth. Run /init inside a session to generate a CLAUDE.md project file.

Claude Code CLI Flags

Session Control	
-c	Continue most recent session
-r <name>	Resume session by name
--resume <id>	Resume session by ID
--bg	Run session in background
Output and Automation	
-p "prompt"	Non-interactive, print and exit
--output-format json	Machine-readable output
--output-format stream-json	Stream JSON as it arrives
--max-turns N	Limit agentic turns
Tools and Permissions	
--allowed-tools	Comma-separated allow list, e.g. Read,Grep
--permission-mode	default | acceptEdits | auto | bypassPermissions
--model <name>	Override model for this session
--dangerously-skip-permissions	**DANGER** No prompts — CI only

Keyboard Shortcuts

Esc Esc	Interrupt current action and rewind the last message
Ctrl+C	Cancel current generation (keeps history)
Shift+Tab	Cycle permission mode: ask → auto → bypass
Ctrl+B	Push session to background; reclaim terminal
@filename	Include a file's contents inline in your message
!`cmd`	Run a bash command and inject output into message
↑ / ↓	Navigate message history
/keybindings	Edit shortcuts — stored in ~/.claude/keybindings.json

Slash Commands

Context management	
/compact	Compress history to free context. Pass a focus hint: /compact keep auth module
/clear	Wipe conversation history entirely. Start fresh.
/status	Show context usage, session ID, model, cost
Session navigation	

By **Dave Child** (DaveChild)
cheatography.com/davechild/
aloneonahill.com

Not published yet.
Last updated 23rd June, 2026.

Cheatography

Claude Code Cheat Sheet

by Dave Child (DaveChild) via cheatography.com/1/cs/48210/

Slash Commands (cont)

/rename	Name the current session (no arg = auto-name)
/fork	Branch conversation at this point (try alternatives without losing current state)
/resume	Resume a previous session by name or ID
/export	Export conversation as plain text

Tools and Config

/model <name>	Switch model mid-session. Takes effect next message.
/doctor	Diagnose auth, config, and connection issues
/init	Generate CLAUDE.md for current project
/schedule	Create a recurring automated task
/help	List all current slash commands

Model Aliases

opus / sonnet / haiku	Short aliases: `/model opus`
auto	Claude picks model per task

Planning and Flow Commands

/plan	Ask Claude to produce a plan before writing any code. Review and approve, then Claude executes. Use for any non-trivial change.
/goal	Set a persistent session goal. Claude references it to stay on track across compacts and model switches. Useful for long sessions.
/effort	Set reasoning depth: low | normal | high. Default on Opus 4.8 is high. Lower for speed/cost, higher for architecture decisions.
/checkpoint	Save current conversation state as a named restore point. Useful before risky tool sequences.
/review	Request a code review of current working state. Add --fix to apply findings immediately.

Planning and Flow Commands (cont)

/workflow	Trigger a dynamic workflow — sequences of steps Claude will walk through. See ~/.claude/commands/ for custom workflows.

Typical Flow Example:

```
/goal Refactor the auth module to PSR-12
/plan
# review the plan, press y to approve
# Claude writes code
/compact focus on auth module
/review --fix
```

Custom Slash Commands

File Locations

.claude/commands/	Project command
~/.claude/commands/	Personal commar

Each .md file becomes a slash command nam
Invoked with /filename.

Example: .claude/commands/pr-review.md

```
Review the current branch diff aga
 List bugs and security issues as
 Then summarise in one sentence.
 Arguments: $ARGUMENTS
```

Use !`git diff main` in the file body to p
Use `@path/ to/file` to inline a file. Use `$A`
runtime args: `/pr-review focus on SQL`.

By **Dave Child** (DaveChild)
cheatography.com/davechild/
aloneonahill.com

Not published yet.
Last updated 23rd June, 2026.

Cheatography

Claude Code Cheat Sheet

by Dave Child (DaveChild) via cheatography.com/1/cs/48210/

CLAUDE.md: Project Instructions

Read at the start of every session. The primary way to give Claude persistent context about your project without repeating yourself.

Locations (in priority order)

./CLAUDE.md	Project root — committed to repo, shared with team
./.claude/CLAUDE.md	Project-local, can be gitignored
~/.claude/CLAUDE.md	Personal global — applies to all projects

What To Put In It

Stack & conventions	PHP version, framework, coding standard (PSR-12)
Build & test commands	How to run tests, linter, deploy
Off-limits files	Files Claude should never touch
Project structure	Where things live and why
Behaviour rules	Always write tests, never edit migrations

Example CLAUDE.md snippet

```
Stack: PHP 8.3, Laravel 11, MySQL 8
Standard: PSR-12. Run: ./vend - or/ bin /pint
Tests: ./vend or/ bin /pest — must pass before commit
Never edit files in /datab - ase /mi gra tions/
```

Headless and Pipe Mode

```
# Pipe content as context
cat src/Au th.php | claude -p "Find security issues "
git diff main | claude -p " - Review this diff"
git log --oneline -10 | claude - p " Write release notes"
# JSON output (includes cost, session ID, turn count)
claude -p "list TODOs" --outp - ut- format json | jq .
# Read-only: no file writes or bash
claude -p " explain archit ect - ure " \
    --a llo wed -tools Read,Grep
# Multi- step: reuse session across calls
SID=$( claude -p " analyse test failur es" \
    --o utp ut- format json | jq -r '.sess ion _id')
claude -p "fix the failur es" -- resume " $SI D"
# Append system prompt for specia lised context
claude -p " review errors " \
    --a ppe nd- sys tem -prompt "You are an SRE expert "
```

Use -p for scripts and CI. Claude runs the prompt and exits — no interactive session.

Permissions and Safety

Permission Modes

default

acceptEdits

auto

bypassPermissions

Tool allow/deny in settings

```
# .claude/settings.json
 {
   " per mis sio ns": {
     " all ow": ["Ba sh(git:)", " te"],
     " den y":  ["Ba sh(rm -rf)"]
   }
 }
```

Restrict in CLI

```
claude --allo wed -tools Read,Grep th"
```

By **Dave Child** (DaveChild)
cheatography.com/davechild/
aloneonahill.com

Not published yet.
Last updated 23rd June, 2026.