



Claude Code Commands

Verified 2026-08-05

A practical guide to commands in Claude Code for interacting with your terminal and codebase.

Contents

Mental model	2
Basic session control	2
Project and files	2
Model and context	3
Background tasks	4
Code review and changes	4
Troubleshooting	5
Further reading	6

Mental model

Claude Code brings Claude directly into your terminal. Commands start with a `/` and provide ways to manage Claude's behavior, handle files, navigate your project, and trigger tools without leaving the chat interface. Think of `/` commands as controlling the shell that surrounds Claude's AI engine.

Basic session control

Task	Command
Start fresh	<code>/clear</code>
View shortcuts	<code>/help</code>
Exit CLI	<code>/exit</code> or <code>/quit</code>

Commands to manage the lifetime of your Claude Code session. Running `/clear` provides an empty context window without losing project memory.

```
/clear my-previous-task
```

Tip: You can resume a previous session with `/resume` and selecting it from the picker.

Project and files

Task	Command
Initialize	<code>/init</code>

Task	Command
Switch dir	<code>/cd <path></code>
Add dir	<code>/add-dir <path></code>

Claude Code needs to know what files it can access. `/init` creates a `CLAUDE.md` to guide Claude in your repository. `/cd` safely changes directories while preserving context.

```
/cd src/components
```

Gotcha: If you just want Claude to read files in another directory without moving the session, use `/add-dir` instead.

Model and context

Task	Command
Change model	<code>/model</code>
Adjust effort	<code>/effort [level]</code>
See context	<code>/context</code>

Large codebases fill the context window fast. `/context` visualizes what is taking up space. `/model` and `/effort` control which model runs and how much reasoning it applies.

```
/effort high
/compact
```

Note: Use `/compact` to summarize the current conversation to free up context space.

Background tasks

Task	Command
List tasks	<code>/tasks</code>
Detach session	<code>/background</code>
Copy to background	<code>/fork</code>

Claude Code can run tasks in the background. `/tasks` lists background work and subagents. `/background` lets the whole session run detached so you can close your terminal.

```
/fork
```

Tip: Use `/subtask <task>` to hand a side task to a background agent that will report back to your current conversation.

Code review and changes

Task	Command
View diff	<code>/diff</code>
Code review	<code>/code-review</code>
Batch changes	<code>/batch <inst></code>

These commands analyze or manipulate your code at scale. `/diff` opens an interactive viewer. `/code-review` checks for correctness and can apply findings with `--fix`.

```
/code-review medium --fix
```

Warning: `/batch` will create isolated git worktrees and open PRs for each unit of work it identifies.

Troubleshooting

Task	Command
Run checkup	<code>/doctor</code>
Fix setup	<code>/doctor</code>
Rewind session	<code>/rewind</code>

If something breaks, `/doctor` runs a setup checkup to diagnose installation, path, and configuration issues. `/rewind` rolls code and conversation back to a checkpoint.

```
/doctor
```

Gotcha: `/debug` enables debug logging mid-session, which is useful if you didn't start Claude with `--debug`.

Further reading

- [Claude Code Interactive Mode](#) — Keyboard shortcuts, Vim mode, and command history.
- [Claude Code Skills](#) — Learn how to create your own custom commands.
- [Claude Code Agent View](#) - Learn about background tasks.