

Claude Code Cheatsheet

Your essential quick reference to master Claude Code

</> CLAUDECODE-LAB

1. Basic Commands

<code>claude</code>	Start interactive mode
<code>claude -p "..."</code>	One-shot execution (non-interactive)
<code>/help</code>	Show help
<code>/clear</code>	Clear context
<code>/compact</code>	Compress context and continue
<code>/cost</code>	Check session cost
<code>/exit</code>	Exit Claude Code

2. Keyboard Shortcuts

<code>Ctrl + C</code>	Cancel current operation
<code>Esc</code>	Cancel input
<code>Esc</code> ×2	Go back to previous message
<code>Shift + Tab</code>	Next suggestion / toggle mode
<code>↑ / ↓</code>	Navigate input history

3. Writing CLAUDE.md

- ▶ **Project overview** — What you're building in 1-2 lines
- ▶ **Tech stack** — Language, framework, DB, key libraries
- ▶ **Coding standards** — Naming, formatter, lint config
- ▶ **Common commands** — `build / test / dev / deploy`
- ▶ **Off-limits areas** — Files/dirs that should not be modified

4. Effective Prompting

- ▶ **Specify exact file names** — "Fix @src/api/user.ts"
- ▶ **State expected output format** — JSON / Markdown / diff etc.
- ▶ **Show Before/After examples** — Providing I/O samples boosts accuracy
- ▶ **Break into steps** — Design → Implement → Test → Refactor
- ▶ **Describe verification** — "npm test should pass"

5. Useful Pipe Integrations

```
git diff | claude -p "Review this diff"
Instant code review
```

```
cat error.log | claude -p "What caused this?"
Root cause analysis from logs
```

```
cat package.json | claude -p "Clean up deps"
Audit package.json dependencies
```

```
git log --oneline -20 | claude -p "Write release notes"
Generate CHANGELOG from commits
```

Pro Tips

- ▶ **@filename** — Add files directly to context
- ▶ **Run /compact regularly** — Maintain accuracy in long sessions
- ▶ **Separate planning & execution** — Plan first, then Act