



Claude Code Cheat Sheet

Essential Commands | MCP Tricks | Pro Tips | Resources

From The Complete Claude Code Skill Stack | 1,200+ Skills | 50+ Categories

FREE PREVIEW

QUICK START Up and running in 60 seconds

```
$ npm install -g @anthropic-ai/claude-code
$ cd my-project && claude
# Set API key (first time only):
$ export ANTHROPIC_API_KEY=sk-ant-...
```

TERMINAL

10 ESSENTIAL COMMANDS

- `claude` Start interactive session in current directory
- `claude -p "prompt"` One-shot mode — ask and exit. Fast & cheap
- `claude /init` Generate CLAUDE.md config for your project
- `claude /compact` Compress context to save tokens & stay sharp
- `claude --resume` Continue where you left off in last session
- `cat file | claude -p "..."` Pipe any file or output directly into Claude
- `claude /cost` Check token usage and costs for this session
- `claude /export` Export conversation as Markdown file
- `claude mcp add <n> <cmd>` Connect MCP server (DB, API, browser, etc.)
- `claude --model sonnet` Use faster/cheaper model for simple tasks

TIP: Use `--allowedTools "Edit,Write,Bash(git*)"` to restrict Claude to safe operations only.

MCP: CONNECT CLAUDE TO EVERYTHING

Model Context Protocol lets Claude talk to databases, APIs, browsers and more.

● PostgreSQL / MySQL

```
claude mcp add db -- npx @anthropic-ai/mcp-postgres $DB_URL
```

Query your database directly. Claude reads schema, writes & runs SQL.

● GitHub

```
claude mcp add github -- npx @modelcontextprotocol/server-github
```

Create issues, review PRs, search repos from Claude.

● Filesystem

```
claude mcp add fs -- npx @modelcontextprotocol/server-filesystem /path
```

Give Claude controlled read/write access to directories.

● Brave Search

```
claude mcp add search -- npx @anthropic-ai/mcp-brave-search
```

Let Claude search the web for docs and solutions in real-time.

● Puppeteer (Browser)

```
claude mcp add browser -- npx @anthropic-ai/mcp-puppeteer
```

Claude can navigate, screenshot, and interact with web pages.

● Memory

```
claude mcp add memory -- npx @modelcontextprotocol/server-memory
```

Give Claude persistent memory across sessions.

TOKEN OPTIMIZATION & COST TRICKS

1. Use /compact aggressively

After 20-30 messages, context bloats. /compact keeps key info, saves 40-60% tokens.

2. One-shot with -p for quick tasks

claude -p "question" costs ~500 tokens vs 2,000+ in interactive mode.

3. Scope requests tightly

"Fix null check in auth.ts line 42" is 10x cheaper than "fix all bugs".

4. CLAUDE.md replaces repeated context

Put stack, rules in CLAUDE.md. Claude reads it automatically — no re-explaining.

5. Batch related tasks together

"Add error handling to all API routes" in one prompt beats 8 separate ones.

ESSENTIAL GITHUB REPOS Bookmark these

anthropics/claude-code

github.com/anthropics/claude-code

Official CLI — issues, docs, changelog.

anthropics/courses

github.com/anthropics/courses

Free courses: prompt engineering, tool use, RAG.

google/generative-ai-docs

github.com/google/generative-ai-docs

Token optimization patterns. Works with any LLM.

modelcontextprotocol/servers

github.com/modelcontextprotocol/servers

Official MCP server directory. 50+ connectors.

punkpeye/awesome-mcp-servers

github.com/punkpeye/awesome-mcp-servers

Community MCP: Slack, Notion, DBs. 200+ tools.

anthropics/anthropic-cookbook

github.com/anthropics/anthropic-cookbook

Production: RAG, agents, structured output, caching.

jxnl/instructor

github.com/jxnl/instructor

Structured output. Typed Pydantic models from Claude.

f/awesome-chatgpt-prompts

github.com/f/awesome-chatgpt-prompts

10k+ starred prompt library. Works with Claude too.

CLAUDE.md STARTER TEMPLATE Copy into your project root

```
# Project: My App
```

CLAUDE.md

```
## Stack
```

```
- Next.js 14, TypeScript, Tailwind, Prisma, PostgreSQL  
- Testing: Vitest + Playwright | CI: GitHub Actions
```

```
## Rules
```

```
- Always use TypeScript strict mode  
- Use server components by default  
- Run `npm test` before suggesting changes are complete  
- NEVER modify .env files or expose secrets
```

```
## Commands
```

```
- Dev: npm run dev | Test: npm test | Build: npm run build
```

PRO TIP: Add multiple CLAUDE.md files in subdirectories for module-specific rules. Claude reads the nearest one automatically. Great for monorepos!

COPY-PASTE WORKFLOW RECIPES

1 Automated Code Review

```
git diff main | claude -p "Review this diff. Flag bugs, security issues"
```

Pipe your PR diff directly for instant review before merging.

2 Fix All Failing Tests

```
claude -p "Run npm test, read failures, fix each one, re-run until pass"
```

Claude iterates: run tests, read errors, fix code, repeat.

3 Generate API Docs

```
claude -p "Read src/routes/ and generate OpenAPI 3.0 spec"
```

Full OpenAPI spec from route handlers. Automatic.

4 Security Audit

```
claude -p "Audit src/ for OWASP Top 10. List vuln with file, line, fix"
```

Systematic security scan with actionable fix suggestions.

5 DB Migration

```
claude -p "Read Prisma schema, generate migration for user roles"
```

Schema-aware migration with best practices built in.

QUICK TIPS THAT SAVE HOURS

- Shift+Tab toggles single-line vs multi-line input mode
- Escape twice cancels a running operation without losing context
- Add # IMPORTANT comments in CLAUDE.md for unbreakable rules
- Run claude /doctor to diagnose connection, auth, and config issues
- Use --allowedTools to restrict Claude to safe operations only
- Pipe: git log --oneline -20 | claude -p "summarize recent work"

This is just 2% of the full stack.

1,200+ skills | 50+ categories | Instant setup | Lifetime access

Get the Complete Stack -> skills.thenichesociety.ro

Launch Price: \$29.99 (Save 70%)