

# CLAUDE CODE CHEAT SHEET

## Getting Started

```
npm install -g anthropic-ai/clause-code
claudie
Start your first session!
```

## Essential Commands

- ▶ `/terminal-setup` - Enable shift-enter
- ▶ `/init` - Create claude.md (# adds memory)
- ▶ `/theme` - Toggle dark/light mode
- ▶ `/install-github-app` - GitHub actions
- ▶ `Shift+Tab` - Auto-accept ("yolo")
- ▶ `Shift+Tab+Tab` - Plan mode
- ▶ `Esc` - Interrupt Claude
- ▶ `Esc+Esc` - History navigation
- ▶ `!` - Bash mode
- ▶ `@` - Mention files/folders
- ▶ `Ctrl+R` - Full output/context
- ▶ `Ctrl+V` - Paste image
- ▶ `/claude-c` or `--resume` - Continue session

**Git Worktrees (Parallel CC Instances)**

```
git worktree add ../feature-a -b feature-a
cd ../feature-a && claude
```

## Custom Commands

**Setup Directories**

```
mkdir -p .claude/commands
mkdir -p ~/.claude/commands
```

(personal commands)

**Create Commands**

```
echo "Analyze performance..." > .claude/commands/optimize.md
echo "Fix issue #$ARGUMENTS..." > .claude/commands/fix-issue.md
```

**Use Commands**

```
/project:optimize
/project:fix-issue 123
```

## Useful Prompts

**Planning & Strategy**

- ▶ "Think deeply about the best approach"
- ▶ "Think harder about edge cases"
- ▶ "Break into tasks and run subagents"

**Analysis**

- ▶ "Look through git history"
- ▶ "Trace login process from front-end to database"

**Design & Testing**

- ▶ "Implement [mock.png]. Screenshot & iterate"
- ▶ "Add tests, run them and fix failures"

**Pull Requests**

- ▶ "Summarize changes to auth module"
- ▶ "Create PR with testing info"

**Extended Thinking**

- ▶ `think` - Basic
- ▶ `think harder/longer` - Deeper
- ▶ `ultrathink` - Maximum

## MCPs (Model Context Protocol)

**Basic Syntax**

```
claude mcp add <name> <command> [args...]
claude mcp add --transport sse <name> <url>
claude mcp add --transport http <name> <url>
```

**Install Popular MCPs**

**context7** (documentation library)

```
claude mcp add --transport http context7 https://mcp.context7.com/mcp
```

**Puppeteer** (browser automation)

```
claude mcp add puppeteer -s user -- npx -y @modelcontextprotocol/server-puppeteer
```

**Playwright** (browser automation)

```
claude mcp add playwright npx @playwright/mcp@latest
```

**Supabase** (database)

```
claude mcp add supabase -s local -e SUPABASE_ACCESS_TOKEN=your_token -- npx -y @supabase/mcp-server-supabase@latest
```

**Linear** (project/issue management)

```
claude mcp add linear -s user -- npx -y mcp-remote https://mcp.linear.app/sse
```

**Sentry** (issue management)

```
claude mcp add --transport http sentry https://mcp.sentry.dev/mcp
```

**Digital Ocean** (cloud hosting)

```
claude mcp add digitalocean-mcp-local -e DIGITALOCEAN_API_TOKEN=YOUR_TOKEN -- npx "@digitalocean/mcp"
```

## Additional Workflows, Tools, and Resources

**Commands (Custom)**

- ▶ `/commit` - Creates git commits using conventional format with emojis
- ▶ `/create-pr` - Streamlines PR creation with branch, commits, and formatting
- ▶ `/fix-github-issue` - Analyzes and fixes GitHub issues automatically
- ▶ `/check` - Performs comprehensive code quality checks
- ▶ `/prime` - Primes Claude with project context and setup
- ▶ `/add-to-changelog` - Automatically updates project changelog
- ▶ `/create-prd` - Generates product requirement documents
- ▶ `/todo` - Manages and tracks project tasks
- ▶ `/mermaid` - Creates Mermaid diagrams from descriptions

**Workflows**

- ▶ **ClaudeLog** - Comprehensive knowledge repository with advanced Claude Code mechanics and best practices
- ▶ **n8n\_agent** - Comprehensive commands for code analysis, QA, design, documentation, and SDLC
- ▶ **Project Bootstrapping and Task Management** - Structured commands for bootstrapping and managing new projects
- ▶ **Simone** - Broader project management system with documents, guidelines, and processes
- ▶ **Slash-commands megalist** - Comprehensive list of 88+ slash-commands for various development tasks

**Tools**

- ▶ **Claude Code Flow** - Code-first orchestration layer for autonomous code optimization across recursive agent cycles
- ▶ **Claude Composer** - Tool that adds small enhancements to Claude Code
- ▶ **Claude Task Master** - Solves context length limitations through specialized context isolation
- ▶ **Claude Hub** - Webhook service connecting Claude Code to GitHub repositories via @mentions
- ▶ **Claude Squad** - Terminal app managing multiple Claude Code and other AI agents in separate workspaces
- ▶ **Container Use** - Containerized deployment and management tools for Claude Code

**Resources**

- ▶ **Anthropic Documentation** - Official Claude Code docs with installation, usage, and API references
- ▶ **Anthropic Quickstarts** - Collection of slash-commands, claude.md files, and other Claude Code resources
- ▶ **Awesome-Claude-Code** - Curated list of commands, files, and workflows for Claude Code

Created by Sean Matthew | [seanmatthew.me](http://seanmatthew.me) | YouTube: [SeanMatthewAI](https://www.youtube.com/seanmatthewAI) | X: [@\\_sean\\_matthew](https://x.com/_sean_matthew)