

QUICK START TIPS

Be specific: Avoid vague requests
Complex tasks: Break into numbered steps
Before editing: Ask to explore/analyze first
Shortcuts: Press ? to view all

PROJECT & FILES

/init Initialize CLAUDE.md project
/add-dir Add working directory
/memory Edit CLAUDE.md memory
/export [file] Export conversation

MODEL & SESSION

/model Switch AI model
/compact [inst] Compact conversation
/resume Resume previous session
/rewind Undo changes/conversation

CODE QUALITY

/review Request code review
/security-review Security audit
/pr-comments View PR feedback

DEV TOOLS

/sandbox Enable sandboxed bash
/hooks Configure tool hooks
/plugin Manage plugins
/mcp Manage MCP servers
/agents Create custom subagents

WORKSPACE

/clear Clear conversation
/context Visualize token usage
/cost Token statistics
/usage Plan limits/rate status
/status Open Status settings
/config Open Config settings
/todos Show pending items

INPUT PREFIXES

PREFIX	ACTION
#	Save memo to CLAUDE.md
/	Slash command
!	Direct bash execution
@	File path autocomplete

MULTILINE INPUT

\+Enter All terminals
Option+Enter macOS default
Shift+Enter After /terminal-setup

CREATING CUSTOM COMMANDS

PROJECT

```
# .claude/commands/optimize.md
Analyze and suggest optimizations:
→ Usage: /optimize
```

PERSONAL

```
# ~/.claude/commands/security.md
Review security vulnerabilities:
→ Usage: /security
```

ARGUMENTS & REFERENCES

FEATURE	SYNTAX	EXAMPLE
All args	<code>\$ARGUMENTS</code>	/fix 123 high → "123 high"
Positional	<code>\$1</code> <code>\$2</code>	\$1="123", \$2="high"
File ref	<code>@path</code>	@src/utils.js → content
Bash exec	<code>!'`cmd`'</code>	!' git status` → output

NAMESPACE (SUBFOLDERS)

```
.claude/commands/frontend/component.md
→ /component (project:frontend)

.claude/commands/backend/migrate.md
→ /migrate (project:backend)
```

FRONTMATTER OPTIONS

```
--- description: Command description (required for auto-invoke)
allowed-tools: Bash(git:*), Read(*) model: claude-3-5-haiku-
20241022 argument-hint: [param1] [param2] disable-model-
invocation: false ---
```

ALLOWED-TOOLS PATTERNS

PATTERN	DESCRIPTION
Bash(git:*)	All git commands
Bash(npm test:*)	npm test only
Read(*)	Allow all file reads
Edit(src/**)	Edit files in src/
Write(*.md)	Write .md files only
WebFetch(*)	Allow all URL fetch
mcp_github_*	All GitHub MCP tools

SLASHCOMMAND TOOL PERMISSIONS

```
# Set in /permissions SlashCommand:commit # exact match
SlashCommand:/review-pr:* # prefix match
```

Char limit: 15,000 (default)
Env var: SLASH_COMMAND_TOOL_CHAR_BUDGET

MCP COMMANDS

```
# Dynamically discovered from MCP servers /mcp_server_prompt
[args] # Examples /mcp_github_create-issue bug report
/mcp_slack_send-message #general /mcp_db_query SELECT * FROM
users /mcp_docker_logs container-name
```

PLUGIN COMMANDS

```
# From marketplace plugins /plugin-name:command-name /command-
name # if unambiguous # Examples /docker:build /docker:push
registry/image /aws:deploy production /k8s:rollout deployment/app
```

SKILLS VS SLASH COMMANDS

SLASH COMMANDS	SKILLS
Simple, frequent tasks	Complex workflows
Single .md file	Multiple files + scripts
Quick prompt execution	Structured knowledge
Args via \$1, \$2	Template system

MODEL OPTIONS

MODEL	TRAITS	USE CASE
haiku	Fast, low cost	Simple tasks
sonnet	Balanced	General tasks
opus	Best performance	Complex analysis

QUICK REFERENCE

Command locations:

- Project: `.claude/commands/*.md`
- Personal: `~/.claude/commands/*.md`

Block auto-invoke:

```
disable-model-invocation: true
```

UI & EDITOR

/vim Toggle vim mode
/statusline Configure status UI
/terminal-setup Install Shift+Enter
/ide Manage IDE integration
/output-style Set output format

SYSTEM & SETUP

/login Switch Anthropic account
/logout Log out
/doctor Check installation
/help Help documentation
/bug Report issue
/permissions Manage permissions
/release-notes View updates
/exit Exit REPL

CLI COMMANDS

COMMAND	DESCRIPTION
claude	Start interactive mode
claude "task"	Run single task
claude -p "query"	Query and exit
claude -c	Continue recent session
claude -r	Resume conversation
claude commit	Generate Git commit

KEYBOARD SHORTCUTS

KEY	ACTION
Ctrl+C	Cancel current task
Ctrl+D	End session
Ctrl+L	Clear screen
Ctrl+R	Search command history
Ctrl+B	Move to background
Tab	Autocomplete / toggle thinking
↑ ↓	Navigate command history
Shift+Tab	Toggle permission mode

INSTALLATION

```
# macOS/Linux (Homebrew) brew install --cask claude-code #
macOS/Linux/WSL (Shell) curl -fsSL https://claude.ai/install.sh |
bash # Windows (PowerShell) irm https://claude.ai/install.ps1 |
iex
```

CHECKPOINTING (REWIND)

ACCESS	METHOD
Esc+Esc	Open rewind menu
/rewind	Checkpoint options

Restore options:

- Conversation only - keep code, restore messages
- Code only - restore files, keep conversation
- Both - restore entire session

Limit: Bash commands (rm, mv) not tracked

BACKGROUND EXECUTION

Ctrl+B: Move current Bash to background
Useful for npm, webpack, tests, long-running tasks
Check output later with BashOutput tool