

# CLAUDE CODE

## Quick Reference Guide

claude.ai/code | All commands, shortcuts & features

### SLASH COMMANDS

<code>/help</code>	Show help and available commands
<code>/clear /reset /new</code>	Clear conversation and free context
<code>/compact [focus]</code>	Compact conversation, optional focus area
<code>/model [name]</code>	Switch model (opus, sonnet, haiku)
<code>/fast [on off]</code>	Toggle fast output mode
<code>/plan</code>	Enter plan mode (read-only exploration)
<code>/init</code>	Initialize project with CLAUDE.md
<code>/memory</code>	Edit CLAUDE.md and auto-memory files
<code>/config /settings</code>	Open settings interface
<code>/permissions</code>	View/update tool permissions
<code>/diff</code>	Open interactive diff viewer
<code>/commit</code>	Stage and commit changes
<code>/review</code>	Review a pull request
<code>/pr-comments [PR]</code>	Fetch GitHub PR comments
<code>/cost</code>	Show token usage statistics
<code>/usage</code>	Show plan limits and rate status
<code>/status</code>	Show version, model, account info
<code>/context</code>	Visualize context usage as grid
<code>/copy</code>	Copy last response to clipboard
<code>/export [file]</code>	Export conversation as text
<code>/resume /continue</code>	Resume a previous session
<code>/fork [name]</code>	Create fork of conversation
<code>/rewind /checkpoint</code>	Rewind to a previous point
<code>/add-dir &lt;path&gt;</code>	Add a working directory
<code>/mcp</code>	Manage MCP server connections
<code>/hooks</code>	Manage hook configurations
<code>/keybindings</code>	Open keybindings config
<code>/vim</code>	Toggle vim editing mode
<code>/theme</code>	Change color theme
<code>/terminal-setup</code>	Configure terminal keybindings
<code>/doctor</code>	Diagnose installation issues
<code>/sandbox</code>	Toggle sandbox mode
<code>/tasks</code>	List background tasks
<code>/stats</code>	Daily usage visualization & streaks
<code>/insights</code>	Generate session analysis report
<code>/exit /quit</code>	Exit Claude Code

## BUNDLED SKILLS (SLASH COMMANDS)

<code>/simplify</code>	Auto-review and fix recent code changes
<code>/batch &lt;instruction&gt;</code>	Large-scale parallel changes across codebase
<code>/debug [description]</code>	Troubleshoot Claude Code session issues
<code>/loop [interval] &lt;prompt&gt;</code>	Run a prompt repeatedly on a schedule
<code>/claude-api</code>	Load Claude API reference material

## KEYBOARD SHORTCUTS

### General Controls

<code>Ctrl+C</code>	Cancel current input or generation
<code>Ctrl+D</code>	Exit Claude Code (EOF)
<code>Ctrl+G</code>	Open prompt in external editor
<code>Ctrl+L</code>	Clear screen (keeps history)
<code>Ctrl+O</code>	Toggle verbose output (see tool usage)
<code>Ctrl+R</code>	Reverse search command history
<code>Ctrl+B</code>	Background running tasks
<code>Ctrl+T</code>	Toggle task list visibility
<code>Ctrl+F</code>	Kill background agents (press twice)
<code>Ctrl+V / Cmd+V</code>	Paste image from clipboard

### Text Editing

<code>Ctrl+K</code>	Delete to end of line
<code>Ctrl+U</code>	Delete entire line
<code>Ctrl+Y</code>	Paste deleted text (from Ctrl+K/U)
<code>Alt+B / Alt+F</code>	Move cursor back/forward one word

### Mode & Model Switching

<code>Shift+Tab / Alt+M</code>	Cycle permission modes (Normal > Auto > Plan)
<code>Alt+P / Option+P</code>	Switch model without clearing prompt
<code>Alt+T / Option+T</code>	Toggle extended thinking
<code>Esc + Esc</code>	Rewind / restore to previous point

### Multiline Input

<code>\ + Enter</code>	Newline (works in all terminals)
<code>Option+Enter</code>	Newline (macOS native)
<code>Shift+Enter</code>	Newline (iTerm2, WezTerm, Ghostty, Kitty)
<code>Ctrl+J</code>	Line feed

## CLI FLAGS & OPTIONS

### Session Control

<code>claude</code>	Start interactive session
<code>claude "query"</code>	Start with initial prompt
<code>claude -p "query"</code>	Print response and exit (headless)
<code>claude -c</code>	Resume most recent conversation
<code>claude -r "name"</code>	Resume specific session by name/ID
<code>claude -w feature-name</code>	Start in isolated git worktree

### Model & Behavior

<code>--model &lt;name&gt;</code>	Set model: opus, sonnet, haiku, or full ID
<code>--fast [on off]</code>	Toggle fast mode
<code>--permission-mode &lt;mode&gt;</code>	plan, auto-accept, or normal
<code>--max-turns &lt;n&gt;</code>	Limit agentic turns (print mode)
<code>--max-budget-usd &lt;n&gt;</code>	Spend limit before stopping (print mode)
<code>--verbose</code>	Show detailed turn-by-turn output

### Context & Input

<code>--add-dir &lt;path...&gt;</code>	Add extra working directories
<code>--append-system-prompt &lt;text&gt;</code>	Add text to system prompt
<code>--system-prompt &lt;text&gt;</code>	Replace entire system prompt
<code>--json-schema &lt;schema&gt;</code>	Get validated JSON output (print mode)

### Output Formats (Print Mode)

<code>--output-format text</code>	Plain text output (default)
<code>--output-format json</code>	JSON output with metadata
<code>--output-format stream-json</code>	Streaming JSON events

### Permissions & Safety

<code>--dangerously-skip-permissions</code>	Skip all permission prompts
<code>--allowedTools "Bash(git *)"</code>	Auto-allow specific tools
<code>--disallowedTools "Bash(curl *)"</code>	Block specific tools

### MCP & Integration

<code>--mcp-config &lt;path&gt;</code>	Load MCP servers from file
<code>--chrome / --no-chrome</code>	Enable/disable Chrome automation
<code>--agent &lt;name&gt;</code>	Use a specific agent type

## SPECIAL SYNTAX

<code>! &lt;command&gt;</code>	Bash mode - execute shell command directly
<code>@file.js</code>	Mention file (includes content in context)
<code>@src/</code>	Mention directory (includes listing)
<code>@server:protocol://path</code>	Reference MCP resource
<code>/</code> at start of input	Command/skill autocomplete
<code>Tab</code>	Accept prompt suggestion
<code>"ultrathink"</code> in prompt	Trigger high-effort reasoning turn

## MCP SUBCOMMANDS

<code>claude mcp list</code>	List all configured MCP servers
<code>claude mcp get &lt;name&gt;</code>	Get server details
<code>claude mcp remove &lt;name&gt;</code>	Remove a server
<code>claude mcp add --transport http &lt;n&gt; &lt;url&gt;</code>	Add HTTP server
<code>claude mcp add --transport stdio &lt;n&gt; -- &lt;cmd&gt;</code>	Add stdio server
<code>claude mcp add-json &lt;name&gt; '&lt;json&gt;'</code>	Add server from JSON
<code>claude mcp add-from-claude-desktop</code>	Import from desktop app
<code>claude mcp serve</code>	Use Claude as an MCP server
<code>--scope user project local</code>	Set server scope
<code>--env KEY=value</code>	Pass env vars to server
<code>--header "Auth: Bearer tok"</code>	Add auth headers

## AUTHENTICATION

<code>claude auth login</code>	Sign in to Anthropic account
<code>claude auth login --sso</code>	Sign in via SSO
<code>claude auth logout</code>	Sign out
<code>claude auth status</code>	Check auth status (JSON)
<code>claude auth status --text</code>	Check auth status (readable)

## ENVIRONMENT VARIABLES

### API & Model

<code>ANTHROPIC_API_KEY</code>	API key for direct access
<code>ANTHROPIC_MODEL</code>	Default model (e.g., claude-opus-4-6)
<code>CLAUDE_CODE_EFFORT_LEVEL</code>	Reasoning depth: low   medium   high
<code>MAX_THINKING_TOKENS</code>	Fixed thinking budget limit

### Behavior

<code>CLAUDE_CODE_MAX_OUTPUT_TOKENS</code>	Max output tokens (up to 64000)
<code>CLAUDE_CODE_SHELL</code>	Override shell detection
<code>CLAUDE_CODE_DISABLE_FAST_MODE</code>	Disable fast mode (set to 1)
<code>CLAUDE_CODE_DISABLE_GIT_INSTRUCTIONS</code>	Remove git context (set to 1)
<code>CLAUDE_CODE_ENABLE_PROMPT_SUGGESTION</code>	Enable/disable suggestions
<code>DISABLE_AUTOUPDATER</code>	Disable auto-updates (set to 1)

### Cloud Providers

<code>CLAUDE_CODE_USE_BEDROCK=true</code>	Use AWS Bedrock backend
<code>CLAUDE_CODE_USE_VERTEX=true</code>	Use Google Vertex AI backend
<code>ANTHROPIC_BEDROCK_REGION</code>	AWS region for Bedrock
<code>ANTHROPIC_VERTEX_PROJECT_ID</code>	GCP project for Vertex

## CONFIGURATION FILES

<code>~/.claude/settings.json</code>	User settings (all projects)
<code>.claude/settings.json</code>	Project settings (shared/committed)
<code>.claude/settings.local.json</code>	Project settings (gitignored)
<code>~/.claude/keybindings.json</code>	Custom keyboard shortcuts
<code>CLAUDE.md / .claude/CLAUDE.md</code>	Project instructions for Claude
<code>.claude/rules/</code>	Rule files (path-specific context)
<code>.mcp.json</code>	Project MCP servers (shared)
<code>~/.claude.json</code>	User/local MCP servers
<code>.claude/hooks.json</code>	Project hooks configuration

<code>~/.claude/hooks.json</code>	User hooks configuration
<code>.claude/skills/&lt;name&gt;/SKILL.md</code>	Project custom skills
<code>~/.claude/skills/&lt;name&gt;/SKILL.md</code>	User custom skills
<code>.claude/agents/&lt;name&gt;/</code>	Project custom agents
<code>~/.claude/agents/&lt;name&gt;/</code>	User custom agents

## PERMISSION MODES

<code>Normal</code>	Asks permission for each tool use
<code>Auto-Accept</code>	Auto-approves configured tools, shows <code>#õ#õ</code>
<code>Plan</code>	Read-only exploration, shows <code>#ø</code> , asks before changes
<code>Shift+Tab to cycle</code>	Normal !' Auto-Accept !' Plan !' Normal

## HOOK EVENTS

<code>PreToolUse</code>	Before any tool executes
<code>PostToolUse</code>	After a tool succeeds
<code>PostToolUseFailure</code>	After a tool fails
<code>UserPromptSubmit</code>	Before processing user prompt
<code>SessionStart</code>	When session initializes
<code>SessionEnd</code>	When session ends
<code>Notification</code>	When Claude needs attention
<code>Stop</code>	When session is stopping
<code>SubagentStart / SubagentStop</code>	Subagent lifecycle events
<code>TaskCompleted</code>	Background task finished
<code>PreCompact</code>	Before context compaction
<code>WorktreeCreate / WorktreeRemove</code>	Worktree lifecycle
<code>ConfigChange</code>	Settings changed

## TIPS & HIDDEN FEATURES

<code>! npm test</code>	Run shell commands directly (bash mode)
<code>/compact focus on auth</code>	Compact with specific focus area
<code>/doctor</code>	Deep diagnostic of your installation
<code>/stats</code>	See your usage streaks & patterns
<code>/insights</code>	AI-generated session analysis
<code>claude mcp serve</code>	Use Claude itself as an MCP server
<code>claude --init-only</code>	Run init hooks then exit immediately
<code>Tab on suggestion</code>	Accept grayed-out prompt suggestion
<code>Ctrl+R then type</code>	Search through prompt history
<code>CLAUDE_CODE_TASK_LIST_ID=x</code>	Named persistent task list
<code>Cmd/Ctrl+Click PR footer</code>	Open PR in browser from status bar