

CLAUDE CODE 快速参考

CLI 命令、工作流、MCP、Hooks、配置

快速上手

安装与启动

```
npm install -g @anthropic-ai/claude-code
claude --version # start interactive session
claude --help # check version
claude update # update to latest
```

身份验证

```
claude login # browser OAuth
export ANTHROPIC_API_KEY="sk-ant-..."
claude logout # clear session
```

Slash 命令

会话命令

```
/help 显示可用命令
/clear 清除对话历史
/compact 压缩上下文以节省 token
/cost 显示 token 用量和费用
/status 显示会话信息和模型
/new 开启新对话
/config 打开或查看配置
/doctor 诊断配置问题
/init 为项目创建 CLAUDE.md
/login 向 Anthropic 认证
/logout 清除认证
```

工作流命令

```
/bug 提交 bug 报告
/review 发起代码 review
/pr 创建或更新 Pull Request
/commit 提交暂存的更改并附消息
```

键盘快捷键

```
Ctrl+C 取消当前生成
Ctrl+D 退出 Claude Code (EOF)
Escape 取消当前输入 / 编辑
Tab 自动补全文件路径和命令
Up/Down 浏览输入历史
```

CLI 参数

常用参数

```
-p, --print 非交互 (无头) 模式
--model 指定模型: `opus`、`sonnet`、`haiku`
--output-format 输出格式: `text`、`json`、`stream-json`
--allowedTools 限制工具: `Edit,Read,Bash`
--max-turns N 限制对话轮数
--debug 开启调试日志
-n, --name 命名会话
```

无头 / 脚本模式

```
claude -p "explain this error" < log.txt
claude -p "list todos" --output-format json
echo "fix typos" | claude -p
```

配置文件

CLAUDE.md 层级

```
./CLAUDE.md 项目级指令 (提交到仓库)
./.claude/settings.json 项目配置 (权限、hooks)
~/.claude/CLAUDE.md 用户全局指令 (所有项目)
~/.claude/settings.json 用户全局设置
~/.claude/projects/*/CLAUDE.md 按项目用户指令 (不提交到仓库)
```

环境变量

```
ANTHROPIC_API_KEY 直接认证的 API 密钥
CLAUDE_MODEL 默认模型覆盖
CLAUDE_CONFIG_DIR 自定义配置目录路径
```

权限

settings.json

```
{
  "permissions": {
    "allow": ["Read", "Glob", "Grep"],
    "deny": ["Bash(rm *)"]
  }
}
```

权限模式

```
default 高风险工具前询问
--dangerously-skip-permissions 允许所有工具 (仅 CLI/脚本使用)
--allowedTools CLI 参数限制工具集
```

MCP 服务

什么是 MCP 服务?

Model Context Protocol 服务通过自定义工具 (数据库、API、服务) 扩展 Claude Code。通过 CLI 或 `mcp.json` 管理。

CLI 管理

```
claude mcp add server-name -- cmd arg1 arg2
claude mcp list
claude mcp remove server-name
```

.mcp.json 配置

```
{
  "mcpServers": {
    "my-db": {
      "command": "python",
      "args": ["-m", "mcp_server_db"],
      "env": { "DB_URL": "${DATABASE_URL}" }
    }
  }
}
```

作用域

```
./mcp.json 项目级 MCP 服务
~/claude/mcp.json 用户全局 MCP 服务
claude mcp add -s user 添加到用户作用域
claude mcp add -s project 添加到项目作用域
```

Hooks

Hook 事件

```
PreToolUse 工具执行前触发
PostToolUse 工具完成后触发
Notification Claude 发送通知时触发
Stop Claude 完成响应时触发
```

settings.json 示例

```
{
  "hooks": {
    "PreToolUse": [
      {
        "matcher": "Bash",
        "hooks": [
          {
            "type": "command",
            "command": "check-command.sh $INPUT"
          }
        ]
      }
    ]
  }
}
```

Hook 行为

```
exit 0 允许工具继续执行
exit 2 阻止工具运行
stdout JSON 反馈给 Claude
```

SDK 与自动化

无头脚本

```
# Single prompt, get JSON output
claude -p "summarize main.py" \
  --output-format json

# Pipe input
git diff | claude -p "review this diff"
```

CI / GitHub Actions

```
- uses: anthropics/claude-code-action@v1
  with:
    claude_args: >
      --max-turns 5
      --model claude-sonnet-4-6
```

使用技巧

在自动化中用 `--max-turns` 控制成本。用 `--output-format json` 程序化解析结果。结合 `--allowedTools` 提升安全性。

模型

选择模型

```
claude --model opus # most capable
claude --model sonnet # balanced (default)
claude --model haiku # fastest, cheapest
```

模型速查

```
opus Claude Opus——最强能力
sonnet Claude Sonnet——默认, 均衡
haiku Claude Haiku——快速且经济
--model full-name 如 `claude-sonnet-4-6`
```

最佳实践

编写 CLAUDE.md

在 CLAUDE.md 中写入项目约定、技术栈、构建命令和测试说明。保持简洁——Claude 每次会话都会读取它。用 `init` 快速生成。

成本控制

```
/cost 查看实时 token 用量
/compact 上下文过大时压缩
--max-tuxns 自动化中限制轮数
sonnet / haiku 简单任务用更便宜的模型
```

高效提示

```
具体描述 "修复 auth.ts:42 的空值检查" > "修复 bug"
提供上下文 引用文件、函数、错误信息
用 @file 直接引用文件: `@src/main.ts`
使用图片 拖入截图提供视觉上下文
迭代优化 继续追问细化; 用 `clear` 重置
```