

# Claude Code 速查表

快速掌握 Claude Code 的必备参考指南

</> CLAUDECODE-LAB

## 1. 基本命令

<code>claude</code>	启动交互模式
<code>claude -p "..."</code>	一次性执行 (非交互)
<code>/help</code>	显示帮助
<code>/clear</code>	清除上下文
<code>/compact</code>	压缩上下文并继续
<code>/cost</code>	查看会话费用
<code>/exit</code>	退出 Claude Code

## 2. 键盘快捷键

<code>Ctrl + C</code>	取消当前操作
<code>Esc</code>	取消输入
<code>Esc</code> x2	返回上一条消息
<code>Shift + Tab</code>	下一个建议/切换模式
<code>↑ / ↓</code>	浏览输入历史

## 3. CLAUDE.md 编写要点

- 项目概述 — 用1-2行描述你在做什么
- 技术栈 — 语言、框架、数据库、主要库
- 编码规范 — 命名、格式化、lint 配置
- 常用命令 — build / test / dev / deploy
- 禁止修改的内容 — 不应被修改的文件/目录

## 4. 高效提示词技巧

- 指定具体文件名 — "修复 @src/api/user.ts"
- 明确期望输出格式 — JSON / Markdown / diff 等
- 展示前后对比示例 — 提供输入输出样本提高准确性
- 分步骤请求 — 设计 → 实现 → 测试 → 重构
- 说明验证方法 — "npm test 应该通过"

## 5. 常用管道集成

```
git diff | claude -p "审查这个diff"
即时代码审查
```

```
cat error.log | claude -p "原因是什么?"
从日志分析根因
```

```
cat package.json | claude -p "整理依赖"
审计 package.json 依赖
```

```
git log --oneline -20 | claude -p "写发布说明"
从提交生成更新日志
```

## 实用技巧

- @文件名 — 直接将文件添加到上下文
- 定期运行 /compact — 在长会话中保持准确性
- 分离规划与执行 — 先计划，再执行

</> CLAUDECODE-LAB

了解更多 → [claudecode-lab.com](https://claudecode-lab.com)

© 2026 claudecode-lab.com