

2026.3.1 codex与claude code安装配置全流程

Codex

预先安装

Node.js

<https://nodejs.org/en/download/>

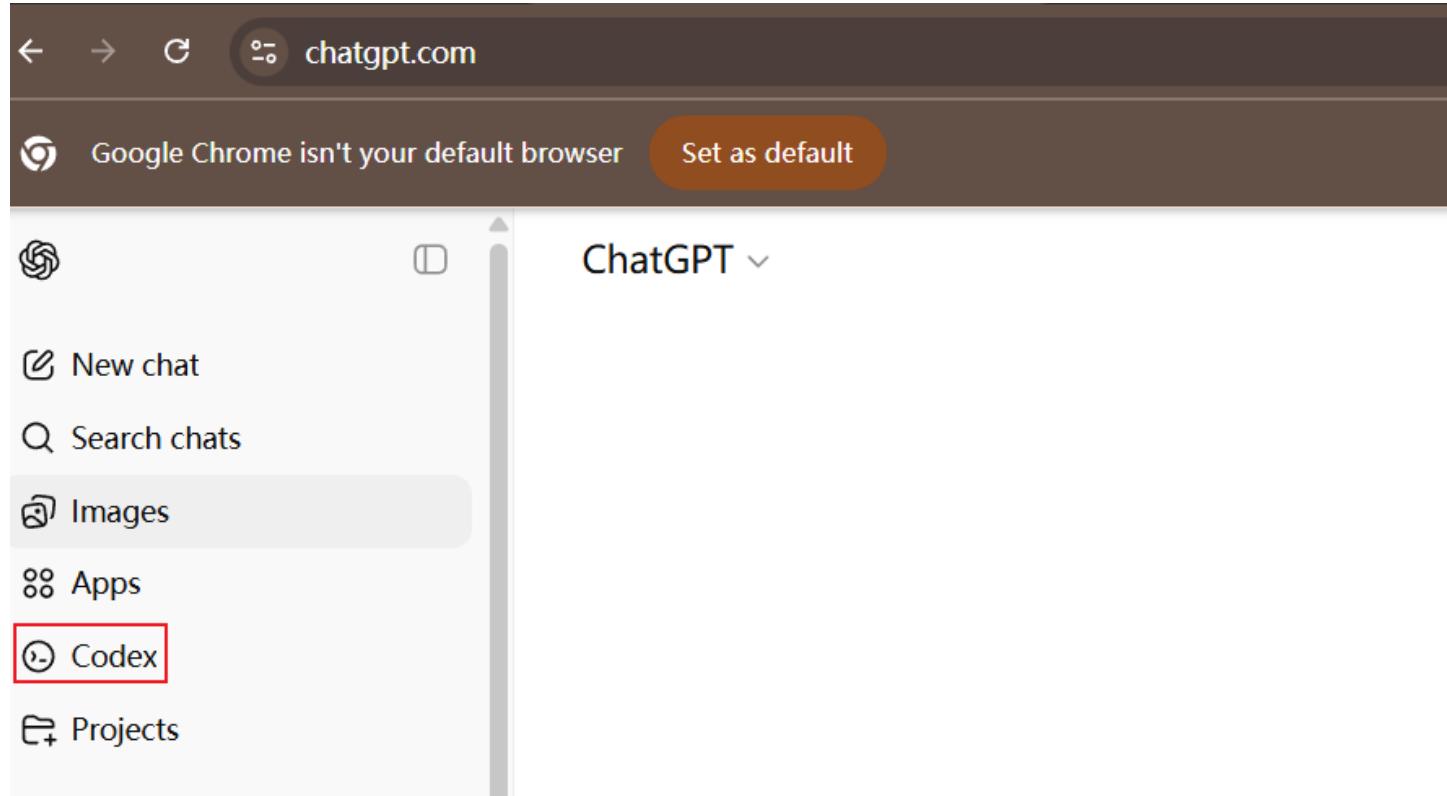
git

<https://git-scm.com/install/windows>

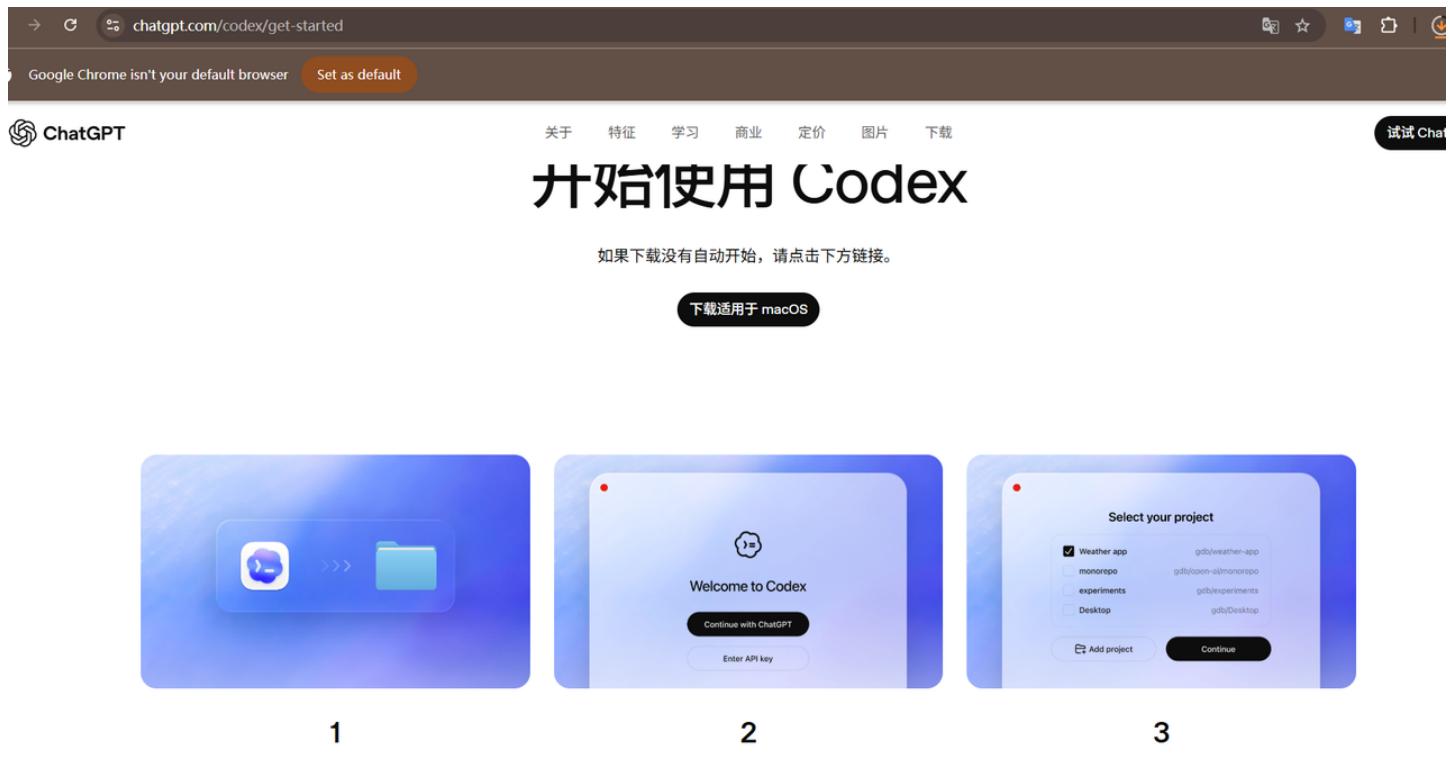
安装命令行

如果有gpt (<https://chatgpt.com/>) 会员，官网上就可以直接使用，windows目前还没有出可以下载到本地的图形界面，mac中有图形界面直接下载安装包就好

windows



macos



1

将 Codex 图标拖到您的“应用程序”文件夹中

2

打开 Codex 并登录。

3

选择计算机上一个可供 Codex 使用的文件夹或 Git 仓库。

windows安装命令行工具

在windows power shell中以命令行的形式运行

代码块

```
1 npm i -g @openai/codex
```

可能会出现这个问题

```
PS C:\windows\system32> npm i -g @openai/codex
npm : 无法加载文件 C:\Program Files\nodejs\npm.ps1，因为在此系统上禁止运行脚本。有关详细信息，请参阅 https://go.microsoft.com/fwlink/?linkID=135170 中的 about_Execution_Policies。
所在位置 行:1 字符: 1
+ npm i -g @openai/codex
+ ~~~
+ CategoryInfo          : SecurityError: () [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
```

右键windows power shell 以管理员身份运行

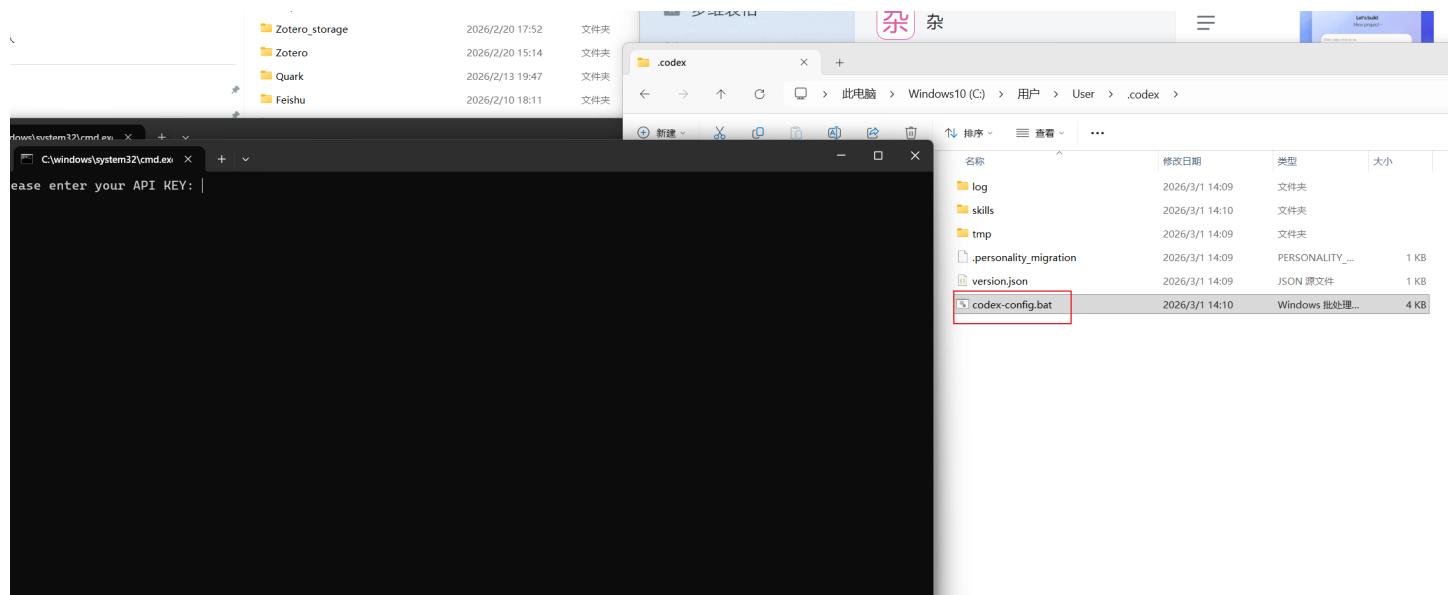
在命令窗口输入，允许运行本地脚本，重新打开输入安装命令即可

代码块

```
1 Set-ExecutionPolicy RemoteSigned
```

如果没有gpt会员的话可以试下这个中转（点击链接即可免费得三天会员），这个中转站的新手指南中可以下载一键配置的命令<https://codex-for.me/?invite=2017>

下载到的脚本放到c/用户/~/codex，双击该命令输入中转站的apikey即可。(该命令实质就是配置模型文件config.toml和在auth.json存入apikey)



进入powershell输入codex，进行基础配置即可。关于sandbox沙箱这个功能尽量选unelevated这个选项，这样风险比较高，但可以调用电脑本地的软件或相关工具）。（如果没注意选了sandbox的话，去c/用户/~/codex目录下的config.toml文件中修改即可。）

```
C: > Users > User > .codex > config.toml
 1 model_provider = "codex-for-me"
 2 model = "gpt-5.3-codex"
 3 model_reasoning_effort = "high"
 4 disable_response_storage = false
 5
 6 [model_providers.codex-for-me]
 7 name = "codex-for-me"
 8 base_url = "https://api-vip.codex-for.me/v1"
 9 wire_api = "responses"
10 requires_openai_auth = true
11
12 [projects.'C:\windows\system32']
13 trust_level = "trusted"
14
15 [windows]
16 sandbox = "unelevated"
```

```
>_ OpenAI Codex (v0.106.0)

model:      gpt-5.3-codex high    /model to change
directory: C:\windows\system32

Tip: New Build faster with Codex.

• Sandbox ready
Codex can now safely edit files and execute commands in your computer

> Write tests for @filename
gpt-5.3-codex high · 100% left · C:\windows\system32
```

codex一些常用命令

/model 选择模型，选择对应模型后有low、medium、high、extra high等几种模式选择，越高的模型思考时间相对较长，token消耗也就越大，按需选择即可

```
Select Model and Effort
Access legacy models by running codex -m <model_name> or in your config.toml

> 1. gpt-5.3-codex (current) Latest frontier agentic coding model.
   2. gpt-5.2-codex           Frontier agentic coding model.
   3. gpt-5.1-codex-max       Codex-optimized flagship for deep and fast
                               reasoning.
   4. gpt-5.2                 Latest frontier model with improvements across
                               knowledge, reasoning and coding
   5. gpt-5.1-codex-mini     Optimized for codex. Cheaper, faster, but less
```

/permissions 选择相关的权限

read only只读不对电脑本地文件做出修改

default只能读这个文件夹里面的一些文件，然后任何的一些修改啊，都会要求来授权

full access为最高的权限，为全自动执行，可以读文件，写文件，也可以访问网络，也可以在这个文件夹之外来做一些事情，**是有风险的**

Update Model Permissions

1. Read Only	Codex can read files in the current workspace. Approval is required to edit files or access the internet.
> 2. Default (non-admin sandbox) (current)	Codex can read and edit files in the current workspace, and run commands. Approval is required to access the internet or edit other files.
3. Full Access	Codex can edit files outside this workspace and access the internet without asking for approval. Exercise caution when using.

/new 开启一个新对话

/init 会创建agent.md文件，在这里你可以写上你的一些要求，每次codex打开都会自动加载，相当于calude code的claude.md的设置。

/compact 压缩对话，将之前的对话进行总结，防止超出上下文限制

/mention @一个文件

/diff 查看当前的这个文件的修改（基于git）

/status 当前状态，可以看到显示了文件夹，然后授权模式，账号的一些信息，用的模型，消耗的token等等

/status

>_ OpenAI Codex (v0.106.0)

Visit <https://chatgpt.com/codex/settings/usage> for up-to-date information on rate limits and credits

Model:	gpt-5.2 (reasoning medium, summaries auto)
Model provider:	codex-for-me - https://api-vip.codex-for.me/v1
Directory:	C:\windows\system32
Permissions:	Default
Agents.md:	C:\Windows\System32\AGENTS.md
Account:	API key configured (run codex login to use ChatGPT)
Collaboration mode:	Default
Session:	019ca81b-13fc-7a63-8283-e6e63aad8fa1
Token usage:	16.5K total (12.7K input + 3.76K output)
Context window:	98% left (18K used / 258K)
Limits:	data not available yet

/mcp 列出装的mcp工具，可以和codex搭配使用，比如

Chrome DevTools MCP，让AI操作浏览器自动化任务，如爬网、测试、页面交互等

/skills 查看现有的skill,在之前的claude code skill中有提到过

/plan 通过该指令进入思考计划模式，先和ai讨论出一个完整可行的计划，再按照这个计划执行会比一直和ai对话让他干活再debug要好的多

对于codex的全局说明

在codex中的CLAUDE.md中进行填写即可（类似claude code的CLAUDE.md）

做一个指令，全局的设置，可以去调教习惯

代码块

- 1 Codex 全局规则
- 2 所有任务执行时必须提供进度反馈，避免用户以为卡死
- 3 称呼规范
- 4 **每次回复时都必须称呼用户为"Minnn"**
- 5 一、操作原则
- 6 优先提交小而可审查的变更。除非明确要求，避免进行大规模重构。
- 7 在编辑之前，先确认需要修改的文件，并用3-6个要点说明修改计划。
- 8 不要凭空编造API、配置或文件路径。如果不确定，先搜索本地文件夹。
- 9 保持修改风格与现有代码风格和架构一致。
- 10 二、安全与密钥
- 11 绝不要将密钥、令牌、私钥、.env 值或凭证粘贴到代码或日志中。
- 12 如果任务需要密钥，请让我通过环境变量提供。
- 13 除非我明确要求，否则不要添加分析统计、遥测或网络请求。
- 14 三、代码质量标准
- 15 如果项目中已有测试，行为发生变化时应新增或更新测试。
- 16 优先保证类型安全，并进行明确的错误处理。
- 17 仅在意图不明显时添加注释。
- 18 四、构建与运行规范
- 19 如果需要运行命令，请给出精确命令，并说明原因。
- 20 当你的修改可能导致构建失败时，先运行最快的相关检查。
- 21 五、输出格式
- 22 对于代码变更：包含简短摘要 + 变更文件列表。
- 23 对于调试：包含假设、已执行的实验步骤，以及最小修复方案。
- 24 六、我的偏好
- 25 我喜欢简洁的解释、明确的步骤，以及可直接复制粘贴的命令。
- 26 默认解释语言：中文。
- 27 七、禁止行为
- 28 长时间沉默不输出任何内容
- 29 只在最后给出结果，中间无反馈
- 30 批量操作时不显示当前处理项
- 31

Claude code

预先安装

Node.js

<https://nodejs.org/en/download/>

git

<https://git-scm.com/install/windows>

命令行安装

在windows powershell中输入下方的命令

代码块

```
1 npm install -g @anthropic-ai/clause-code
```

其他版本的见官网<https://claude.com/product/claude-code>

选择对应版本，复制命令一键安装即可

命令行运行claude即可启动

claude的文件的目录位置

c/用户/./claude



注：如果安装完之后通过命令行无法识别“claude”这个指令，说明没有添加环境变量，需要按照如下方法操作：

1. Win + R 打开运行，输入 `sysdm.cpl` 回车
2. 点击 "高级" 选项卡 → "环境变量"
3. 在 "用户变量" 区域找到 `Path` → 编辑

4. 点击 新建 → 输入 C:\Users\【你的用户名】\.local\bin

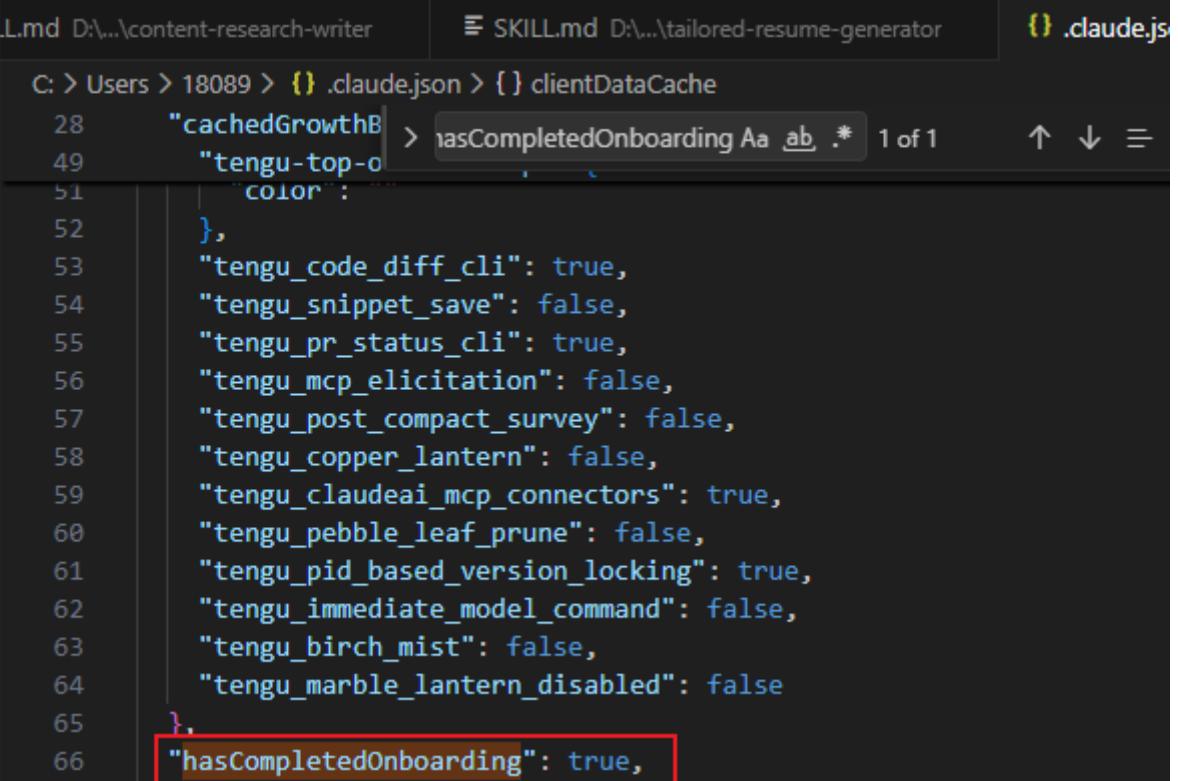
5. 点击 确定 保存所有窗口

6. 重启 PowerShell (完全关闭再打开)

设置不用claude官方登录

进入.claude文件夹加入setting.json进行设置

.claude.json文件中加入参数: "hasCompletedOnboarding": true



```
C: > Users > 18089 > {} .claude.json > {} clientDataCache
28     "cachedGrowthB
29     "tengu-top-o
30     "color":
31 },
32 "tengu_code_diff_cli": true,
33 "tengu_snippet_save": false,
34 "tengu_pr_status_cli": true,
35 "tengu_mcp_elicitation": false,
36 "tengu_post_compact_survey": false,
37 "tengu_copper_lantern": false,
38 "tengu_claudeai_mcp_connectors": true,
39 "tengu_pebble_leaf_prune": false,
40 "tengu_pid_based_version_locking": true,
41 "tengu_immediate_model_command": false,
42 "tengu_birch_mist": false,
43 "tengu_marble_lantern_disabled": false
44 },
45 "hasCompletedOnboarding": true,
```

设置相应的api

现在主要用的是一个claude的中转站<https://foxcode.rjj.cc/auth/register?aff=BD4VK>

里面有详细的配置教程[教程-ClaudeCode·CodeX·Gemini三合一](#)

创建(如果不存在)或编辑 C:\Users\{用户名}.claude\settings.json，输入以下值并保存

代码块

```
1  {
2      "env": {
3          "ANTHROPIC_AUTH_TOKEN": "替换为您的API Key",
4          "ANTHROPIC_BASE_URL": "https://code.newcli.com/clause",
5          "CLAUDE_CODE_DISABLE_NONESSENTIAL_TRAFFIC": 1
6      },
7      "permissions": {
8          "allow": [] ,
```

```
9     "deny": []
10    }
11 }
```

还有一些其他厂商的api可以按找自己的需要去订阅配置，每个厂商都会有相应的配置说明文档的
以下是我之前订阅过的一些厂商的模型

MinMax

🎁 MinMax 跨年福利来袭！邀好友享 Coding Plan 双重好礼，助力开发体验！

好友立享 9折 专属优惠 + Builder 权益，你赢返利 + 社区特权！

👉 立即参与：<https://platform.minimaxi.com/subscribe/coding-plan?code=5mmFTFlPUq&source=link>

智谱GLM

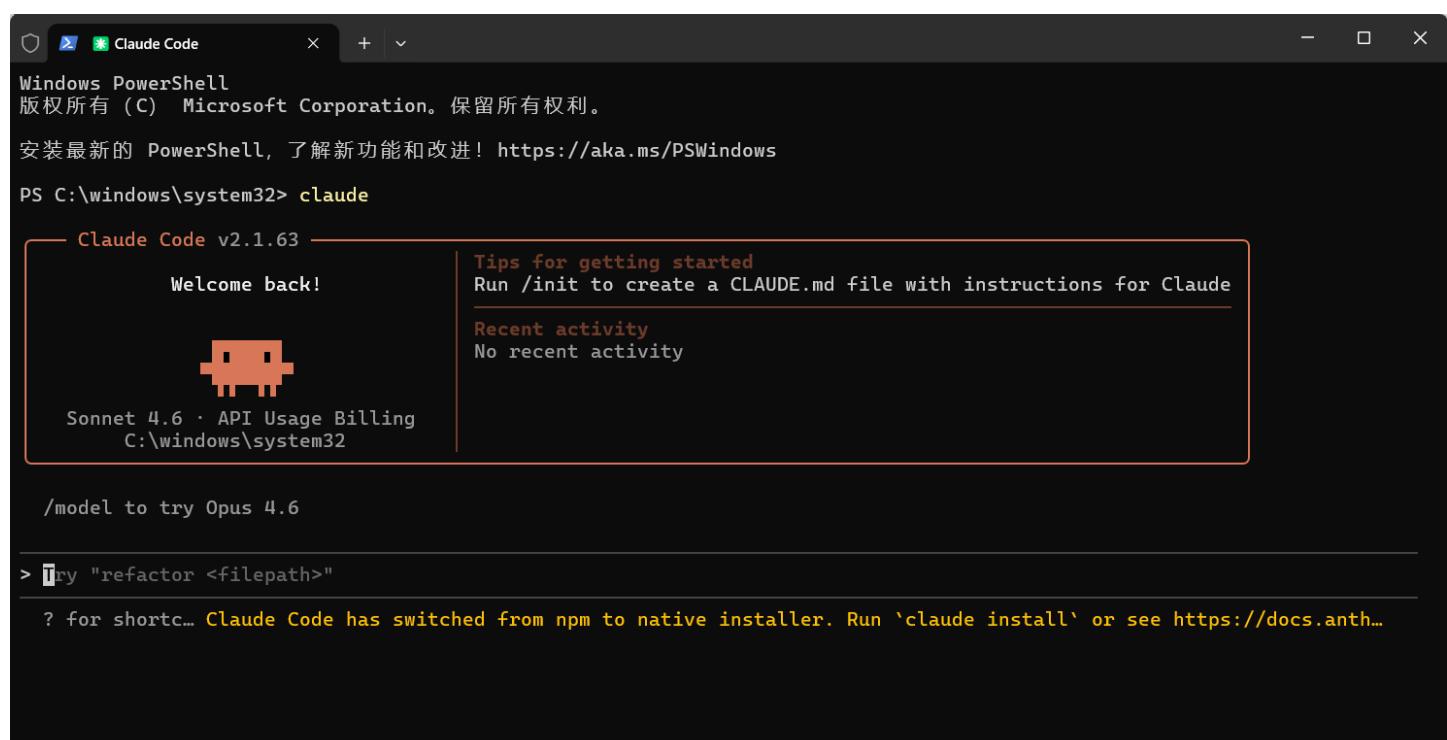
🚀 速来拼好模，智谱 GLM Coding 超值订阅，邀你一起薅羊毛！Claude Code、Cline 等 20+ 大编程工具无缝支持，“码力”全开，越拼越爽！立即开拼，享限时惊喜价！

链接：<https://www.bigmodel.cn/glm-coding?ic=9QZHBJ8CLM>

火山

方舟 Coding Plan 支持 Doubao、GLM、DeepSeek、Kimi 等模型，工具不限，现在订阅折上9折，低至8.9元，订阅越多越划算！立即订阅：<https://volcengine.com/L/eJPLd-4MLKY/> 邀请码：MXWB66R3

命令行终端运行claude即可



```
Claude Code v2.1.63
Welcome back!
Sonnet 4.6 · API Usage Billing
C:\windows\system32

Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。
安装最新的 PowerShell，了解新功能和改进！https://aka.ms/PSWindows

PS C:\windows\system32> claude

Claude Code v2.1.63
Welcome back!
Sonnet 4.6 · API Usage Billing
C:\windows\system32

Tips for getting started
Run /init to create a CLAUDE.md file with instructions for Claude

Recent activity
No recent activity

/model to try Opus 4.6

> Try "refactor <filepath>"?
? for shortc... Claude Code has switched from npm to native installer. Run 'claude install' or see https://docs.anth...
```

Claude code的一些常用命令

`Ctrl+g` 在记事本中编辑聊天内容

`Ctrl+b` 后台运行任务

`/task` 查看当前运行任务

`Ctrl +c` 退出claude

`/clear`：清除上下文记忆（节省 Token，开启新任务时推荐）。

`/compact`：压缩上下文（处理长任务时使用）。

`/compact` 重点保留用户的需求

`/help`：查看所有可用命令。

`Alt + Enter` (Win) / `Opt + Enter` (Mac)：在对话框中换行。

`Alt + P` 切换模型

`claude update` 强制更新Claude Code版本

`/rewind` 回退或两下esc

`/insights` 回顾你过去一个月的使用记录并总结反馈

`claude --dangerously-skip-permissions` 让claude自主接管任务，无需手动确认

`/resume` 回到之前的对话

`claud -c` 恢复上次的对话

`! ls` 查看当前文件夹下的文件

`/hooks` 在使用工具后执行相关功能（hook逻辑），Add new matcher, Write|Edit,add new hook

`jq -r '.tool_input.file_path' |xargs prettier --write`

选择project settings 所有使用这个项目的人都可以使用这个hook

补充：api配置

如果你有多个厂商的coding plan计划，可以使用下面的工具或alias功能去配置可以参考

1.开源cc-switch直接切换

https://www.bilibili.com/video/BV13Dy6B2EzL/?spm_id_from=333.337.search-card.all.click&vd_source=1883ace75f3996b45995309c86acd625

2.Alias 别名功能（在用 比较简洁）

见之前的笔记，无需安装，一连串命令用一个缩写代替

https://www.bilibili.com/video/BV1hY62BmE7t/?spm_id_from=333.337.search-card.all.click&vd_source=1883ace75f3996b45995309c86acd625

关于 `dangerously-skip-permissions` 可以直接写入别名函数中，让claude自主接管任务，无需手动确认,全自动化，对于那种已经确定的、日常经常用到的工作流可以直接全自动运行

示例

1.claude中转站的配置

代码块

```
1  function cc {
2      $env:ANTHROPIC_BASE_URL = "https://code.newcli.com/clause/droid"
3      $env:ANTHROPIC_AUTH_TOKEN = "输入相关的api密钥"
4      $env:API_TIMEOUT_MS = "600000"
5      $env:CLAUDE_CODE_DISABLE_NONESSENTIAL_TRAFFIC = "1"
6
7      $env:ANTHROPIC_MODEL = "claude-opus-4-6"
8      $env:ANTHROPIC_SMALL_FAST_MODEL = "claude-opus-4-6"
9      $env:ANTHROPIC_DEFAULT SONNET_MODEL = "claude-opus-4-6"
10     $env:ANTHROPIC_DEFAULT_OPUS_MODEL = "claude-opus-4-6"
11     $env:ANTHROPIC_DEFAULT_HAIKU_MODEL = "claude-opus-4-6"
12     clause @args
13 }
```

`dangerously-skip-permissions` 可以直接写入别名函数中，让claude自主接管任务，无需手动确认,全自动化

代码块

```
1  function ccd {
2      $env:ANTHROPIC_BASE_URL = "https://code.newcli.com/clause/droid"
3      $env:ANTHROPIC_AUTH_TOKEN = "apikey"
4      $env:API_TIMEOUT_MS = "600000"
5      $env:CLAUDE_CODE_DISABLE_NONESSENTIAL_TRAFFIC = "1"
6      $env:ANTHROPIC_MODEL = "claude-opus-4-6"
7      $env:ANTHROPIC_SMALL_FAST_MODEL = "claude-opus-4-6"
8      $env:ANTHROPIC_DEFAULT SONNET_MODEL = "claude-opus-4-6"
9      $env:ANTHROPIC_DEFAULT_OPUS_MODEL = "claude-opus-4-6"
10     $env:ANTHROPIC_DEFAULT_HAIKU_MODEL = "claude-opus-4-6"
11     clause --dangerously-skip-permissions @args
12 }
```

`Accept edits on` 只是在写入文件时不用再询问，执行终端命令时还会询问

`clause --dangerously-skip-permissions` 让claude自主接管任务，无需手动确认,全自动化

之前xhs刷到有大佬做的api中转各平台的对比大家也可以参考：<https://www.helpaio.com/transit>，如果有觉得好用的欢迎分享到评论区