

Codex 零基础实战教程，穷爆了！带你速通 15 种玩法

大家好，我是程序员鱼皮。<https://mp.weixin.qq.com/s/Zlkom9qPhTxnW4WcroCxmg>

AI 编程工具这东西吧，真的是风水轮流转。最开始大家在吹 Cursor，后来 Claude Code 大火，而这次轮到了 Codex。

Codex 桌面 APP 不仅能帮你写代码，还能直接操作你电脑上的文件、浏览器、甚至是桌面应用，用好之后甚至能顶一个团队！



本期我会从 Codex 安装到项目实战，从基础功能到核心特性，用 10 多个实战案例，手把手教你玩转 Codex。

不管你是想用它来编程开发、办公提效、还是搞一些好玩的创作，看完这篇都能直接上手。

干货密集，建议先收藏，找个安静的地方慢慢食用~

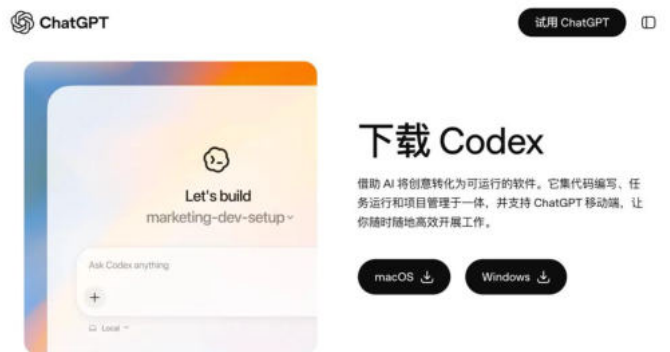


安装和上手

想要使用 Codex APP，你只需要准备一个 ChatGPT 账号。可以免费体验，但是有条件的话最好开个 Plus 会员（每月 20 刀，约 150 元），额度更充裕，足够日常使用了。

有了账号，直接去官网下载安装 Codex APP 就好：

<https://chatgpt.com/zh-Hans-CN/codex/get-started/>

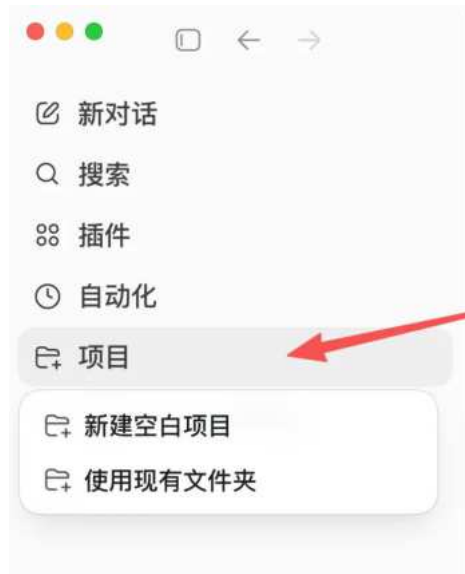


目前支持 macOS 和 Windows (Linux 用户可以使用命令行版本的 Codex CLI)。

下载并安装后，登录你的 ChatGPT 账号，打开之后的界面长这样：

文件操作实战 - 分析磁盘空间

点击左侧的项目入口，选择一个本地文件夹。你可以把它理解为给 AI 划定了一个工作空间，AI 能在这个范围内读取和操作文件。



比如我选择了我的下载文件夹，里面有一堆不知道什么时候存下来的大文件。

在对话框底部，你可以看到权限模式的选项。这里有 3 个选择：

默认权限：AI 可以读取和编辑工作区的文件，需要额外权限时会问你

自动审查：AI 自动帮你审查操作

完全访问：AI 想干啥干啥，不会弹确认框

新手建议直接选「自动审查」，既省时又省心。



选好权限后，在对话框里输入提示词：

帮我分析这个文件夹的空间占用情况

找出所有超过 500MB 的大文件，逐个分析

最后按大小排序列出来，并给出清理建议

然后你就能看到 AI 开始工作了。它会自动执行终端命令来扫描文件，分析每个文件的名称和大小。



左侧是各种面板入口，包括对话管理、插件、自动化等功能。中间就是对话框，你跟 AI 所有的交互都在这里完成。

跟平时用的 AI 聊天工具差不多，还挺清爽的，不像传统 IDE 那么复杂，上手零门槛。

接下来我们直接开始用。

基本使用

直接开聊

安装好之后，跟你用过的其他 AI 工具一样，在对话框里输入内容发送就能开始聊天了。

适合处理一些简单的日常工作，比如帮你查资料、总结内容、规划方案。

我让它帮我查一下今天有什么 AI 编程方面的热点：

今天有哪些值得关注的 AI 编程热点？



Codex 会自动联网搜索最新信息，然后帮忙整理总结，这下不怕错过新资讯了。



这只是开胃菜，Codex 真正强大的能力是操作你本地的文件和电脑，下面我们试试。



最后给你一份清晰的报告，列出哪些大文件占了多少空间，并给出清理建议。



我主打一个听话，让 AI 帮我删除没用的预览文件：

删除预览文件

成功帮我节省了 6.8G 的空间，效果不错吧~



但如果我让 AI 删除一个工作区外的文件呢？会发生什么？

试试看，在当前工作空间，开启一个新对话，比如我让 AI：

帮我删除「鱼皮新书出版」目录下的所有文件

可以直接把这个目录拖拽到对话框中，然后执行任务：

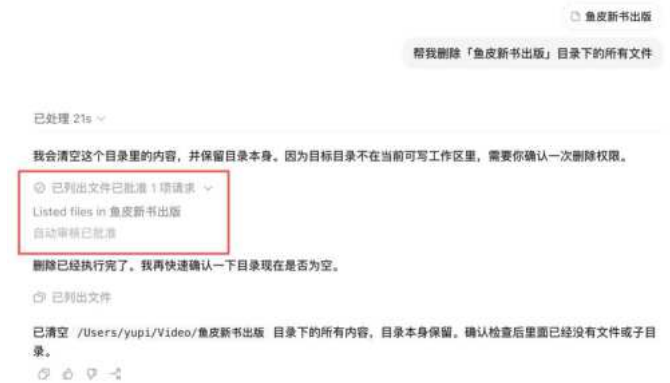
我们该在鱼皮的下载区中做什么？



如果你选择的权限模式是「默认模式」，AI 想删除文件的时候会弹出一个确认框，问你同不同意，防止 AI 搞乱你的电脑。



但由于我选的是「自动审核」，AI 自己完成了审核并批准，省去了人工操作的麻烦。



这样一来，你完全可以吧 Codex 当成你的私人文件管理助手，分析空间、清理垃圾、批量重命名，以前要手动折腾半天的事情，现在一句话就搞定了，是不是挺方便的？

查看使用情况

做完第一个任务，大家肯定很关心消耗了多少 tokens 吧？

来，一起看看用了多少额度。

点击左下角的「设置」，点击「剩余额度」，就能看到你 5 小时内还剩多少配额、本周还剩多少比例，什么时候刷新。

Codex 的额度按 5 小时和 1 周来限制，Plus 用户的额度还是比较大方的，日常使用完全够。



你也可以在对话框里输入 `/状态`，这种以 `/` 开头的快捷输入叫做「斜杠命令」，是

Codex 内置的快捷操作方式：



输入之后，Codex 会直接在对话中显示当前的上下文余量和额度信息。



OK，到这里你已经体验了 Codex 最基础的能力：对话 + 操作本地文件。恭喜，已经超过了 60% 的同学！

接下来我们提升一下难度，用 Codex 做一个完整的网站项目。过程中你会接触到不少 Codex 的核心用法，包括计划模式、AI 生图、浏览器预览、批注修改等等。

AI Agent 编程实战

在开始之前，先进入设置，把工作模式从「适用于日常工作」切换为「适用于编程」，这样 AI 的回复会更专业、更适合开发场景。



项目介绍

我要用 Codex 来为自己定制一个专属的「电子名片」。

把信息告诉 AI，它帮忙生成一个精美的网站。

还能用 AI 生图能力生成个性化的头像插画，不用自己去找素材。

做完之后发给别人，他们打开链接就能看到你的信息了。

第一步、计划模式，先讨论再动手

新建一个项目文件夹（比如叫 `namecard`），在 Codex 中打开。

模型选择目前最新的 GPT-5.5，速度选「标准」就够了，智能程度选「高」。

权限我直接给了完全访问，省得它每次操作都来找我确认，反正大部分情况下我都是无脑点同意的。

最关键的是，点击对话框左下角的 `+` 号，把「计划模式」打开。计划模式下，AI 不会直接开始写代码，而是先帮你规划方案、问你细节，确认没问题了再动手。



输入下面这段提示词，启动！

帮我生成个人电子名片网站，以下是我的信息：

1. 姓名：程序员鱼皮

2. 职业：程序员 / AI 编程博主

3. 简介：前腾讯全栈开发，现自主创业，带团队开发了编程导航、面试帮、鱼皮 AI 导航等产品

4. 联系方式：GitHub: liyupi，网站: codefather.cn

要求：

1. 帮我生成一张卡通风格的程序员头像插画

2. 提供多种风格主题可切换

3. 响应式布局，手机也能正常显示

4. 界面要有设计感，参考苹果官网的简洁风格

AI 会先自己思考，然后可能会追问你一些细节，在弹出的问题面板中选择就好。

已处理 1m 1s

我先问一下当前项目结构，确认是空目录，已有前端项目，还是需要从零搭起来；同时看一下生成头像资产该放进哪种技术栈最顺手。

已探索 1 次搜索 1 个列表已运行 1 条命令

这里基本是一个只有 Git 元数据的空目录，所以计划会按“从零创建静态电子名片站”来定。头像插画这块我会按专用的 imagegen 技能来规划，避免把视觉资产和页面实现割裂开。

已探索 1 个文件 1 个列表已运行 2 条命令

我还看到仓库没有任何提交、也没有已跟踪文件；.DS_Store 是现有未跟踪文件，后续计划里会避开它。现在唯一会影响成品观感的偏好是头像是否追求“本人相似度”。

正在询问 问题

头像插画希望采用哪种方向？

1. 品牌化插画 (推荐)
2. 等你给照片
3. 极简符号头像
4. 否，请告知 Codex 如何调整

ESC 提交

最后会生成一份实现计划文档，包括简介、核心要求、测试计划等信息。你做的网站越复杂，越要仔细阅读这个文档。

我还看到仓库没有任何提交、也没有已跟踪文件；.DS_Store 是现有未跟踪文件，后续计划里会避开它。现在唯一会影响成品观感的偏好是头像是否追求“本人相似度”。

已询问 1 个问题

个人电子名片网站实现计划

Summary

从空目录创建一个零依赖静态单页网站：index.html、styles.css、script.js 和 assets/avatar-programmer.png。页面首屏就是电子名片本体，采用苹果官网式的克制留白、系统字体、细腻阴影和高质感排版。

Key Changes

- 生成一张“品牌化卡通程序员头像”：友好的中文 AI 编程博主形象，不追求真人照片相似度。放入 assets/avatar-programmer.png 并在页面中使用。
- 名片内容包含：程序员鱼皮、程序员 / AI 编程博主、简介、GitHub /liyupi、网站 codefather.cn。
- 提供 4 个主题切换：洁白、深色、科技蓝、活力。用 variables 控制背景、文字、圆角色、边框和阴影。

实施此计划？

1. 是，实施此计划
2. 否，请告知 Codex 如何调整

ESC 提交

第二步、观察 Agent 自主执行

如果没有问题，就确认方案，然后 AI 就开始自主干活了。

它先是使用内置的 Image Gen 技能生成了卡通头像文件：

实施计划

已处理 1m 19s

我会按这个计划直接落地：先生成项目可用的头像资产，再创建纯静态页面、主题样式和本地持久化脚本，最后做一次本地检查。这里会用 imagegen 技能生成卡通头像。

头像已经生成，整体很适合做首屏名片视觉：干净、友好、偏高级插画。我现在把它落到项目的 assets 目录里，然后补齐 HTML/CSS/JS。

正在运行 git status --short



然后编写代码，一次性生成了多个文件：

已运行 mkdir -p assets && cp ./Users/yupi/codex/generated_images/019e3e20-b935-7132-8281-ae660d...

头像资产已经复制进项目了。接下来我会一次性写入三个静态文件：页面结构、四套主题样式、以及主题切换脚本。

正在创建 styles.css +392 -0

已创建 index.html +76 -0

已创建 script.js +31 -0

已创建 styles.css +392 -0

页面代码和样式已经写好了。现在我会做几项验证：静态文件是否完整、链接是否正确、主题持久化是否工作，以及几个常见视口下有没有横向溢出。



写好代码后还会检查代码、自主打开浏览器测试验证，还考虑到了一些容错：

生成电子名片网站



浏览器验证已通过。桌面和手机看起来也干净，没有横向溢出。脚本我再手动加一层 localStorage 容错，让页面在隐私模式或存储受限时也不会坏。

正在编辑 script.js +18 -2

容错补好了。我再跑一遍浏览器验证，确保改动没有影响主题状态和响应式表现。



过了 7 分多钟，AI 完成了整个任务，全程不需要你手动操作~

已处理 7m 33s

已实现完成：创建了纯静态电子名片站点，包含 index.html、styles.css、script.js，并把生成的卡通程序员头像放到了 assets/avatar-programmer.png。

可以直接打开 index.html 预览，无需启动服务。页面包含 4 套主题切换、localStorage 持久化、响应式布局，以及 GitHub / 网站链接。

验证已完成：用 Playwright 检查了桌面、平板、手机视口，无横向溢出；头像加载正常；4 个主题均可切换；刷新后主题保持；链接分别指向 https://github.com/liyupi 和 https://codefather.cn/，.DS_Store 保持未动。



avatar-programmer.png 图像 - PNG 打开

已编辑 3 个文件 +517 -2 撤销 审核

我们可以看到所有生成的文件，点击可以查看文件里的代码：

已编辑 3 个文件 +517 -2 撤销 审核

index.html +76 -0

```
1 <!DOCTYPE html>
2 <html lang="zh-CN">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6   <meta
7     name="description"
8     content="程序员鱼皮的个人电子名片：程序员 / AI 编程博主，前腾讯全栈开发，现自主创业。"
9   />
10  <title>程序员鱼皮 | 电子名片</title>
11  <link rel="preload" href="assets/avatar-programmer.png" as="image" />
12  <link rel="stylesheet" href="styles.css" />
```

script.js +31 -0

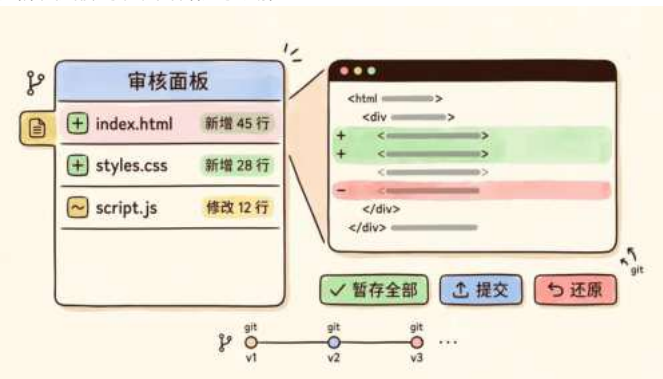
styles.css +392 -0

还可以点击「审核」，打开侧边栏的审查面板，查看本次变更的所有文件：



Codex 底层用的是 Git (一种记录代码变更历史的工具) 来管理所有文件的改动。你能看到每个文件新增了什么、删除了什么, 还可以灵活地应用和撤销代码, 后面进阶功能部分我会详细给大家讲解。

当然, 你也可以直接让 AI 帮你运行网站。AI 会执行终端命令, 启动开发服务器来运行网站。



点击访问地址后, Codex 会在右侧面板打开内置浏览器, 便于你来预览效果。

第三步、查看效果 + 迭代修改

我们做的是纯前端静态网站, 找到生成的 index.html 文件, 右键在浏览器中打开, 就能直接看到效果了。



如果某个地方不满意, 你可以点击浏览器右上角的「批注」按钮, 然后直接在页面上选中要修改的元素, 写上你的修改意见, 发送给 AI。

PC 端的效果我觉得还不错, 布局清晰、主题切换也很丝滑。



而且它还自动兼容了移动端的展示, 手机上打开排版也是正常的。

AI 会自动定位到对应的代码并精准修改，不用你自己去翻代码找位置。改完之后，刷新页面就能看到效果了：



方便是方便，不过这个速度真不敢恭维。。。

做完后我们再看一眼使用情况，这次完整项目消耗了多少额度，还好还好~



总结一下，要用 AI 开发一个网站，我们只要把需求告诉 AI、确认方案、等它自己写完并测试就好，中间几乎不用手动操作。

恭喜，看到这里你已经超过了 70% 的同学！

到这里你已经能用 Codex 从零开发一个网站、预览效果、按需修改了。接下来带大家看看它还有哪些核心功能，学完之后你不仅能做网站，还能让 AI 帮你操控浏览器抓数据、定时自动执行任务、甚至操控整套电脑。

核心功能详解 - 常用功能

接下来我把 Codex 的功能分成「常用功能」和「进阶功能」两大块来讲。

常用功能是日常会高频使用的，进阶功能则是有点入门槛、但掌握后能让你效率翻倍的。

4.1 插件系统

在左侧的「插件」面板里，你能看到 Codex 的插件市场。

Codex 内置了不少精选插件，比如 Computer Use 操控电脑、Chrome 操控浏览器、Spreadsheets 表格处理、Presentations 演示文稿，这些是 OpenAI 官方提供的核心能力。此外还有大量编程类和工具类插件，涵盖网站部署、游戏开发、对接 GitHub 等各种场景，你可以在插件市场里按需搜索和安装。



比如我们来安装 Netlify 插件。Netlify 是一个免费的网站托管服务，装好插件之后，一句话就能把你做的网站部署上线，让别人都能访问。



点击安装 Netlify 插件，同意后会自动弹出浏览器，使用 GitHub 等方式登录

Netlify，一步步完成授权，最后 Codex 成功安装并连接了 Netlify。



然后我们用 Netlify 来部署之前开发完成的电子名片网站，在对话中通过

@Netlify 调用插件。

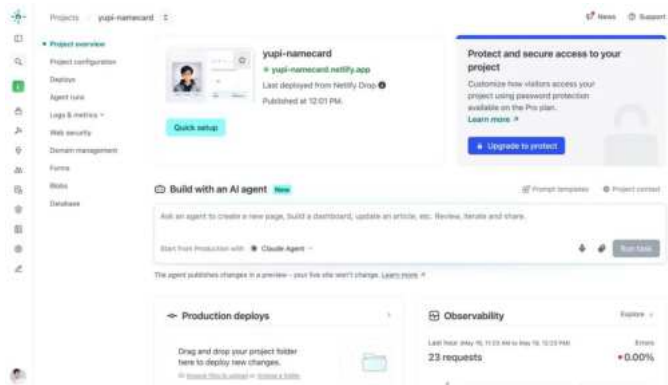
执行过程中，AI 会找我们确认，并自动创建一个新的 Netlify 项目来部署网站：



搞定！以后我想展示自己的信息，直接用这个链接就行了：



还可以打开 Netlify 后台对项目进行管理：



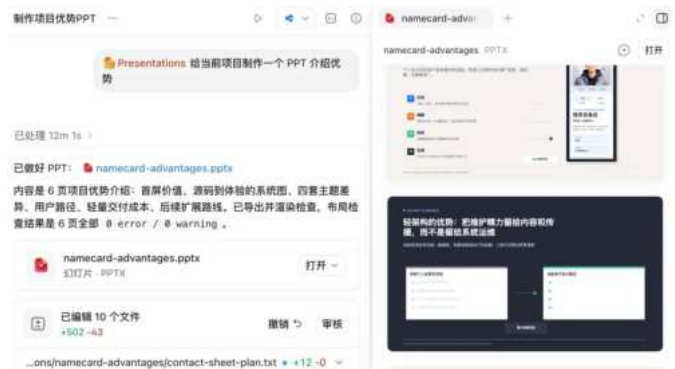
你可以在 Codex 右上方的侧边栏中整体查看当前项目的概览信息，包括后台任务、打开的浏览器、使用的插件等：



点击后台任务，还能看到具体的终端日志，记录了网站服务器什么时候收到了请求、请求了什么资源等信息：



同样的，如果你要处理 Excel 表格、做 PPT，使用对应的插件让 AI 帮你就好，生成的文件还能在侧边栏直接预览：



4.2 Browser Use

前面我们用的内置浏览器可以预览页面、做批注和修改，但如果想让 AI 真正去操控浏览器、自动点击、填表单、翻页，就需要 Browser Use 浏览器操作功能了。

使用 Browser Use

进入设置 → 浏览器，确保 Browser Use 功能已开启。你还可以在这里设置权限规则和禁止打开的域名：



在对话中通过 @浏览器 来调用这个能力。比如我让 AI 帮我打开一个网站并截图：

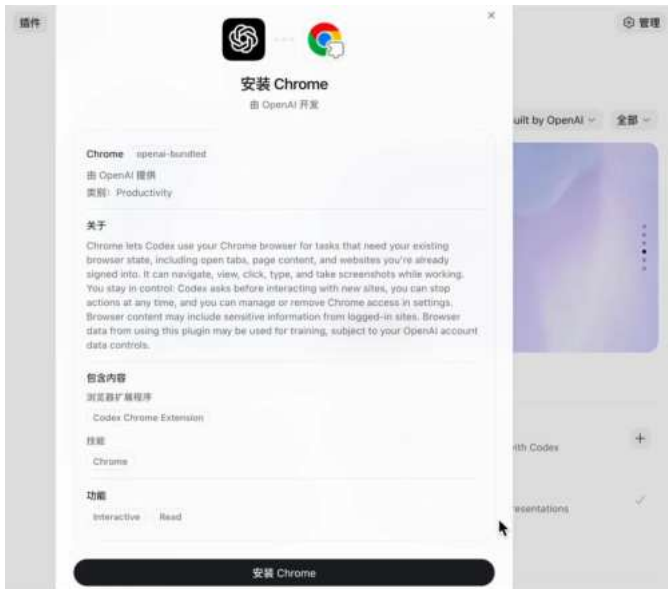
@浏览器 帮我打开鱼皮的面试刷题网站 mianshiya.com，找到 AI 相关的面试题库并截图

可以看到 AI 打开了浏览器，从主页找到了 AI 大模型面试题库，然后进入了详情页，成功完成了截图。不过有时候操作不太稳定，多试几次就好。



使用 Chrome 扩展

此外，你还可以安装 Codex 的 Chrome 扩展插件，它可以操控你电脑上已经登录的 Chrome 浏览器。好处是能保留你的登录状态，而且可以在后台执行不占屏幕。适合需要登录网站才能进行的操作，比如帮我在自己的后台批量分析和数据。



使用前需要先在你的 Chrome 浏览器中安装 Codex 扩展程序，跟着 Codex 的指引操作就好：



安装好之后，我让 AI 帮忙从我已经登录的面试鸭网站上抓取一些数据：

Chrome 获取我在面试鸭 mianshiya.com 最新收藏的 5 个题目信息，并汇总成表格

可以看到 AI 不仅连接了我电脑上的 Chrome 浏览器，而且还识别到了我已经打开的标签页，会控制这些标签页，跳转到我的个人主页并抓取最新的 5 条数据：

最终抓取到的数据表格非常清晰，链接前面甚至还有图标~



4.3 Computer Use

如果说 Browser Use 只能操控浏览器，那 Computer Use 就是让 AI 操控你的整个电脑。AI 能看到你的屏幕内容、移动鼠标、点击按钮、打字，还能操作微信、飞书等任何桌面应用。

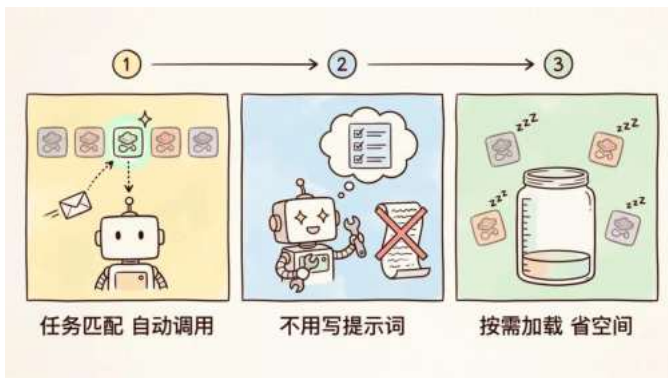
进入设置 → 电脑操控，安装 Computer Use 插件。



你可以在这里看到所有已连接的应用，比如前面装的 Chrome 扩展，其实也属于 Computer Use 体系的一部分。



我们来体验一下。用 @电脑 来调用，让 AI 帮我看看当前的桌面壁纸，然后生成一张类似风格的新壁纸：



进入左侧的「插件」面板，切换到「技能」Tab 页，在这里能够可视化安装和管理技能。

Codex 自带了几个内置的技能，比如 Image Gen 图片生成、OpenAI Docs 查官方文档、Skill Installer 安装社区技能、Skill Creator 创建新技能、Plugin Creator 创建插件等。



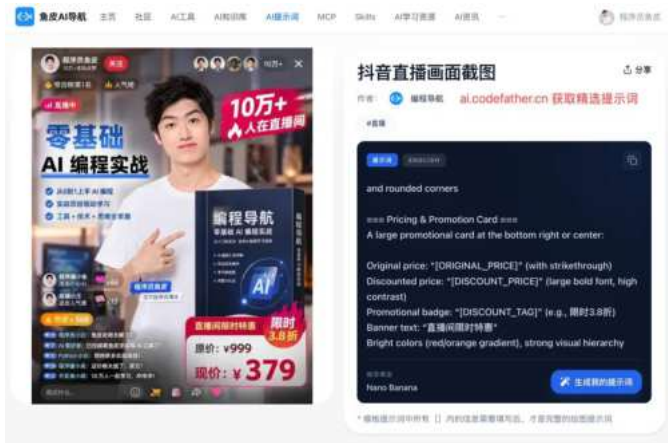
下面我先带大家用内置的图片生成技能，然后安装社区里别人做的技能，最后再来自己创建一个技能。

使用图像生成技能

先来看内置的 Image Gen 图像生成技能。

前面做电子名片的时候我们已经用它生成了卡通头像。除了头像之外，你还可以用它来做 UI 素材、海报、横幅、表情包等等。

这里我想生成一个有趣的图片，让鱼皮直播卖鱼皮，可以到 鱼皮 AI 导航网站 上找到 AI 生图的提示词模板，然后复制提示词模板。



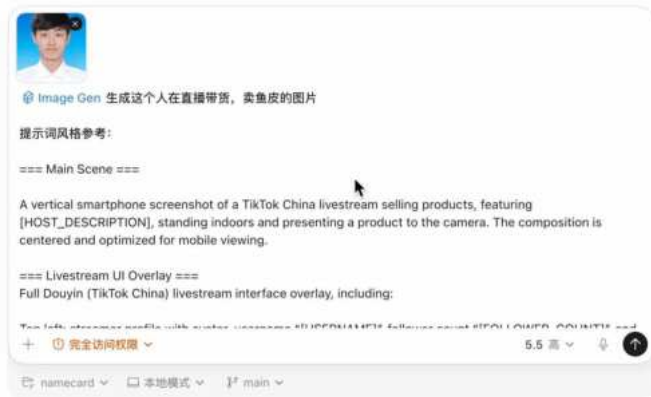
然后在 Codex 对话框中输入 \$ 符号 + 技能名称，可以快速调用技能。并且把提示词模板、鱼皮的照片都提供给 AI：

\$image-gen 生成这个人在直播带货，卖鱼皮的图片

提示词风格参考：

@从鱼皮 AI 导航拿到提示词模板

要在 namecard 中构建什么？



看看 AI 生成的图片，你觉得怎么样？是不是夸爆了？



不过图片生成比普通对话消耗的额度更多，用的时候留意一下剩余配额。

使用社区技能

内置的技能数量有限，还有很多宝藏在社区里。

比如我自己经常用的几个技能：用于联网搜索的 Firecrawl、用于获取最新技术文档的 Context7、用于美化前端页面的 UI UX Pro Max 等等。

另外，我之前写过一篇优质 AI 编程扩展大全，推荐了几十个好用的技能和 MCP，可以到我免费开源的《AI 编程零基础教程》阅读：



接下来我带大家实际安装一个社区技能，顺便整个活儿，让 AI 制作一个苹果风格的快闪动画视频。

首先，要安装动画制作技能 remotion-best-practices。可以使用

skill-installer 技能，它能帮我们快速安装其他技能。

对了，安装社区技能时 **要注意安全**，由于我这里安装的是知名技能，就直接输入技能名称让 Codex 帮我安装了。如果你想安装不太知名的技能，更稳妥的方式是把技能的 GitHub 链接发给 Codex，让它帮你检查后再精准安装。

Skill Installer remotion-best-practices

已处理 2m 54s >

Installed remotion-best-practices successfully at:

/Users/yupi/.codex/skills/remotion-best-practices

I verified its SKILL.md declares name: remotion-best-practices. Restart Codex to pick up new skills.

📄 🔄 🗑️ 🚧

安装完成后，在技能管理面板里就能看到新增的技能了：



然后我们使用技能，让 AI 制作动画：

\$remotion-best-practices 帮我制作一个苹果风格的快闪动画

文案：帮我做个老脸找观众点赞的故事

搭配有节奏感的纯音 BGM

最终直接给我提供视频文件，必须自主完成任务

AI 会安装制作动画所需的项目和依赖包，然后生成视频和音频，还会渲染单帧，检查画面有没有问题：

📁 已列出文件已运行 1 条命令

依赖安装还在跑，Remotion 包会稍微重一点。趁它继续安装，我先读取默认组件，准备把时间轴、文字节奏和画面语

🔍 已探索 4 个文件 3 次搜索

我现在开始改文件：视频会做成 1080x1920 竖屏短片，节奏大概 9 秒，黑白银灰为主、干净大字、快速闪切；音频用

📄 已创建 1 个文件已编辑 4 个文件已运行 1 条命令

画面和音频脚本已经写进去。现在我会先生成 BGM，再跑类型检查，接着渲染单帧看布局有没有明显问题。

📁 已运行 3 条命令



最后 AI 生成了可直接播放的视频，来看看效果：

已完成，视频文件在这里：



成片规格：1080x1920 竖屏，约 9 秒，已带无歌词电子节奏 BGM。工程也保留在 /Users/yupi/Documents/namecard/apple-flash-video，最终 MP4 是

额，他好像理解错了我文案的意思哈哈，翻车了翻车了。。。

我感觉这种方式更适合做产品发布宣传片、知识点快闪卡、节日祝福视频这类节奏紧凑的短片。

而且我根本没认真写提示词好吧！

你可以指定时长、指定文案、利用生图技能搭配素材，增加更多交互动画等等，感兴趣的同学自己玩一玩。

创建自己的技能

除了用别人的技能，你还可以把自己常用的工作流程封装成技能，以后遇到类似任务一键复用。

技能的本质就是一个 SKILL.md 描述文件，加上一些配套脚本和参考资料。

SKILL.md 里要写清楚这个技能是做什么的、什么时候触发、具体执行步骤，AI 读取后就该知道该怎么干活。



创建技能的最佳方式是，先把一个任务流程跑通一遍，觉得满意之后，使用内置的 `$skill-creator` 技能，告诉 Codex 你的技能要做什么、什么时候触发、有什么细节需要注意，它就会帮你自动生成完整的技能文件。

实操一下，比如我们前面让 AI 生成了直播带货的图片，效果还不错，那就把这个流程封装成一个「直播带货图片技能」：

`$skill-creator` 帮我把上述工作封装为「直播带货图片生成技能」

交互式引导用户输入信息，并生成图片

执行成功后，AI 不仅创建了技能文件，还贴心地教你怎么调用：

@ Skill Creator 帮我把上述工作封装为「直播带货图片生成技能」交互式引导用户输入信息，并生成图片

已处理 1m 42s

已封装好「直播带货图片生成技能」，放在自动发现目录：

@ livestream-commerce-image

我创建了：

- @ SKILL.md：交互式收集人物参考图、商品、价格、直播间 UI 信息，并调用图片生成
- @ prompt-blueprint.md：可复用的直播带货图片提示词模板
- @ openai.yaml：技能列表展示信息

验证已通过：Skill is valid!

以后可以这样调用：

```
test
[livestream-commerce-image] 用这张人物照生成直播带货图，商品是鱼皮，主打香脆、低脂、下酒零食
```

SKILL.md
文档 - MD

打开

之后，使用这个技能，只需要提供一张人物照或者商品照，不用再自己填写又臭又长的提示词模板了，非常方便~



直播带货图片生成

已处理 1m 46s

我会用 `livestream-commerce-image` 技能来做：先读取提示蓝图，再加载你给的头像路径作为主播身份参考，把这张企鹅小盆栽当作商品参考来生成竖屏直播间图。

已探索 2 个文件

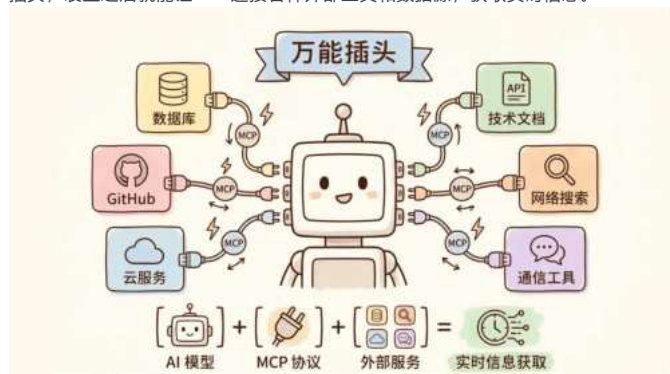
蓝图确认了：它偏向真人主播加商品展示。我会把头像文件作为主播参考，把你贴的企鹅造型草头娃娃/小盆栽做成手持和桌面双展示，价格和直播间文案用默认值补齐。

看看效果，AI 自己识别到我提供的是商品图，也是精准复刻我之前提供的提示词模板：



4.5 MCP 连接外部服务

MCP (Model Context Protocol) 是一个开放协议，你可以把它理解为 AI 的万能插头，装上之后就能让 AI 连接各种外部工具和数据来源，获取实时信息。



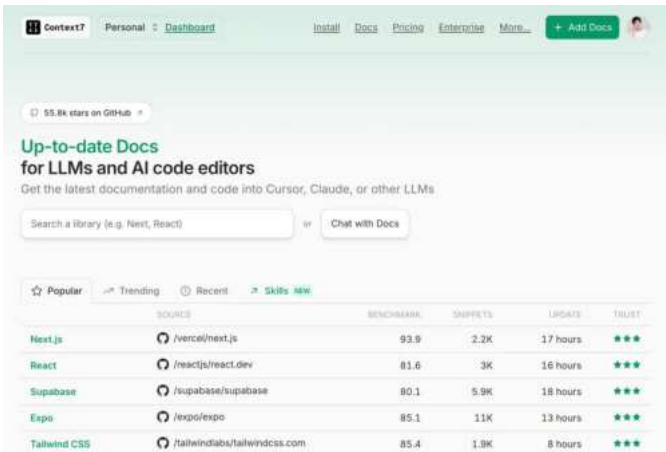
进入设置 → MCP 服务器，可以在这里添加和管理 MCP 服务。

点击「添加服务器」后，你需要手动填写服务器的配置参数才能完成添加，对新手不太友好，我最讨厌填表单了！



好在大部分时候，用前面讲的 Skills 就能解决 MCP 能做的事情，而且 Skills 的安装和使用体验更好。

此外，很多主流的扩展都提供了快捷安装 MCP 的命令，不用你自己手动填参数。比如我带大家安装 Context7，它是一个可以实时获取最新技术文档的服务，开发网站的时候用它来查 API 文档特别方便。



在 Codex 右上角可以打开集成终端，在终端里输入一行命令就能安装：

```
npx ctx7 setup
```

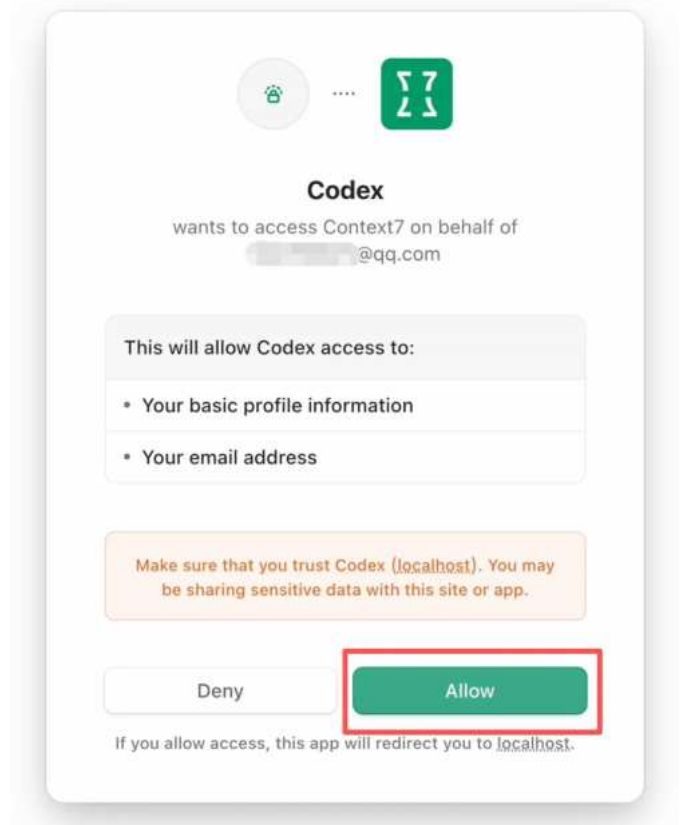
选择安装 MCP server、为 Codex 安装，就搞定了。



安装后重启 Codex，在设置里的 MCP 服务器列表中能看到它。首次使用前，还需要进行身份验证，



在自动弹出的 Context7 网页中登录，然后同意授权。



授权成功后，就可以愉快地使用 MCP 了：



之后开发网站，尤其是需要集成 AI 能力的网站，都可以用 Context7 来获取最新的技术文档。你还可以把它当成一个学习辅助，让 AI 基于官方文档帮你讲解知识。比如我用它来做 OpenClaw 小龙虾的学习助手，这也是利用 AI 学习的一种方式：

```
$context7-mcp 你是我的 OpenClaw 学习助手
```

```
帮我获取最新文档并理解，之后能快速回答我的问题
```

可以看到，AI 查询并获取到了 OpenClaw 官方最新文档库：

Context7 MCP 你是我的 OpenClaw 学习助手 帮我获取最新文档并理解，之后能快速回答我的问题

已处理 28s

我会按你指定的 context7-mcp 流程来做：先把 OpenClaw 在 Context7 里解析成文档 ID，再抓取最新文档要点，之后我就以这些文档为基准回答你的 OpenClaw 问题。

Resolve library id

Context7 找到了官方库 /openclaw/openclaw，而且有多个版本。我会优先看最新列出的 v2026.4.9，先建立一份“学习助手速查底座”：概念、安装、配置、消息平台、Agent/Skills，常见命令和排障入口。

已使用 Context7

当然可以。我已经用 Context7 拉取了 OpenClaw 官方最新文档库：

/openclaw/openclaw/v2026.4.9

我之后会以这个版本文档为基准回答你的 OpenClaw 问题。当前我已经掌握的速查重点是：

• 安装：npm install -g openclaw@latest 或官方安装脚本

然后我们来问 AI 一个问题：

OpenClaw 无法运行，怎么办？

AI 基于官方文档很快给出了精准的排障步骤，这样学习和解决问题又快又准，再也不用查到过时的资料了。

OpenClaw 无法运行，怎么办？

已处理 22s

可以，先别急着重装。按 OpenClaw v2026.4.9 官方排障文档，建议从这组命令开始：

```
bash
openclaw status
openclaw gateway status
openclaw logs --follow
openclaw doctor
```

如果 doctor 发现问题，可以尝试自动修复：

```
bash
openclaw doctor --repair
```

如果你是在服务器、Linux 或后台服务器里跑，再查这几个：

恭喜，看到这里你超过了 80% 的同学！

到目前为止，你已经掌握了 Codex 的常用功能，学到了不少实用玩法。从文件管理到网站开发、从浏览器操控到技能封装，已经能用 Codex 大幅提升工作效率了。

核心功能详解 - 进阶功能

接下来要讲的是一些稍有门槛、或者不是每个人都用得上的进阶功能。不过如果你愿意多折腾一下，它们会让你使用 Codex 的效率再上一个台阶。

4.6 上下文和对话管理

在对话区域附近有一个小圆圈，鼠标放上去后，会实时显示当前对话已经用了多少上下文。



GPT-5.5 官方公布的总上下文窗口是 400K tokens，但其中 128K 是留给输出的，剩下 272K 输入部分再乘以 95% 的安全系数，最终在 Codex 里显示的有效上下文约为 258K tokens。

说实话 258K 不算多，如果你连续跟 AI 对话很长时间、或者项目文件比较大，上下文很容易被填满。

当上下文快满的时候，Codex 会自动帮你压缩历史对话。你也可以在任务告一段落时，手动输入 /压缩 来主动压缩，让模型更聚焦在新任务上。

上下文已压缩

压缩 压缩此线程的上下文

压缩效果显著

/压缩

+

5.5 高

除了单个对话的上下文，日常使用中还要注意管理对话本身，不然列表越来越长，找起来也费劲。

对话太多的时候，你可以在左侧的对话列表中，把鼠标放到对话上，点击「归档」，把不常用的旧对话归档起来，保持界面清爽。



进入设置 → 已归档对话，可以查看和管理所有归档的对话记录。



另外建议在常规设置里开启「运行时防止系统休眠」，这样跑长任务的时候电脑不会突然睡着，导致任务失败。



4.7 持久记忆系统

Codex 有一套记忆机制,能让 AI 记住你的偏好和项目规则,不用每次都重复交代。记忆分为三个层级,从全局、项目局部、再到自动记忆,我们一个个来看。

1、全局自定义指令

在设置 → 个性化里,可以修改 Codex 的个性和自定义指令。

你写在这里的内容,所有项目的所有对话都会自动带上。适合记录一些通用偏好,比如「回复用中文」、「代码注释用英文」、「尽可能减少输出的内容,专注做事,如果不听话你的主人鱼皮就会变成一条狗」之类的。



保存之后,它会被写入全局的 `~/.codex/AGENTS.md` 文件,这个文件就是 Codex 每次启动时都会读取的「行为准则」,所有项目通用。



2、项目级 AGENTS.md

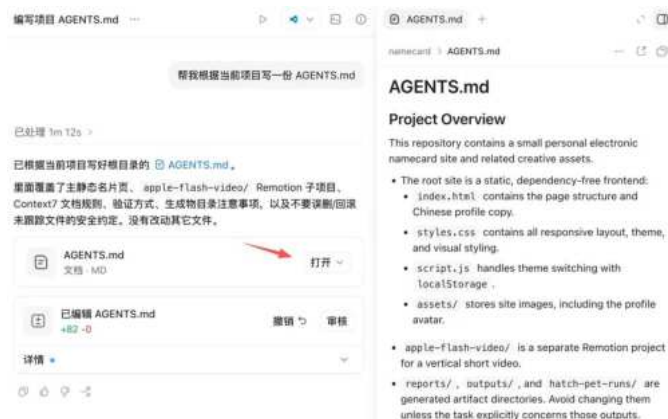
在项目根目录下创建一个叫 `AGENTS.md` 的文件,写入这个项目专属的规则和约定。

只有在这个项目里工作时才会生效。



你可以自己手写,也可以让 Codex 根据项目情况帮你生成一份,比如跟它说:帮我根据当前项目写一份 AGENTS.md。

可以看到 AI 生成了一份非常详细的 AGENTS.md 文档,包括项目总览、一些规范等等。



3、自动记忆

在设置 → 个性化中手动开启自动记忆。

开启后,AI 会在对话空闲一段时间后,自动在后台总结出有用的信息存为记忆,后续遇到相关场景时会自动召回,让 AI 越用越懂你。



不过太短的对话它不会记,额度快用完的时候也不会触发记忆生成。

4.8 定时自动化

Codex 支持定时任务功能。

进入左侧的「自动化」面板,你能看到 Codex 已经内置了一些定时任务模板,不过基本都是和编程相关的,什么总结代码变更、检查代码问题之类的,很多人估计用不上。我们不妨来自己新建一个更实用的自动化任务。



创建定时任务的方式有 2 种。

1、手动创建任务

一种是在自动化面板里点「新建」，比如我让 AI 帮我搜集每日热点，需要填写任务名称、提示词（帮你做什么）、触发时间、模型、推理程度。

运行环境选「本地」就好，意思是让 AI 直接在你当前电脑上执行任务，不需要额外隔离的工作环境。

标题：每日热点搜索

提示词：从国内外搜集今日 AI 相关热点，整理成 HTML 结构的报告



创建成功后，时间一到 Codex 就会自动开启一个对话来执行任务，我们也可以先手动执行一次看看效果。



点击任务，能够查看详细信息：



点击某个运行历史记录后，还能查看正在执行的任务对话，建议多观察任务的表现，持续迭代优化提示词：



2、让 AI 创建任务

另一种更自然的方式是，直接在对话里跟 Codex 说，让 AI 帮你创建任务。

比如我是一名内容创作者，每天都要截大量的图片，时间一长，文件夹里全是看不懂的文件名，找图的时候巨痛苦。



所以我让 Codex 帮我自动整理，先选择项目，然后输入提示词：

帮我创建一个自动化任务

每小时扫描一次「鱼皮的图片库」中最近 3 小时的图片文件

并根据图片内容自动完善图片的中文名称

很快，AI 就自动帮你建好了，可以点击查看创建出来的任务信息，它的提示词比我们提供的更完善了，并自动选择了模型。



我们手动执行任务来测试一下，效果还不错吧，AI 会根据图片内容，自动给文件起一个能看懂的名字。


```

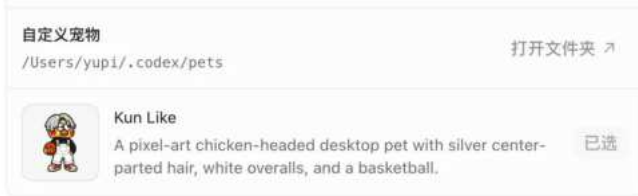
npx petdex@late +
+ namecard git:(main) * npx petdex@latest install kun-like
Need to install the following packages:
petdex@0.4.0
Ok to proceed? (y) y

petdex collects anonymous usage stats (install volume, OS, agents wired up).
No personal data, no file contents. Disable any time:
petdex telemetry off
Details: https://petdex.crafter.run/legal/telemetry

|
v
Downloading kun-like.

```

安装成功后，进入外观界面，选中刚刚安装的宠物。



然后我们回到 codex 主页，使用 /宠物 指令唤醒。我脑袋中已经响起那个熟悉的 BGM 了（噫），你听到了么？

要在 namecard 中构建什么？



3. 自己创建宠物

你还可以通过 Codex 内置的 \$hatch-pet 技能自定义生成宠物，上传照片或者文字描述都行。

比如我把自己的头做成宠物。AI 会先分析上传的图片，给宠物起个名字。



```

已处理 34s

I'll use the hatch-pet workflow you pointed at and keep a visible checklist as it moves from setup to the final packaged pet. I'll first read the local skill files and prepare a run folder around your penguin reference.

已探索 2 个文件已运行 1 条命令

The local scripts are present, and the reference is a 597x598 JPEG. I'm going to name the pet "Yupi Penguin" and keep the toy-like penguin traits from the image: black body, white belly, blue scarf, yellow beak/feet, round eyes, and the tan tuft on top.

正在运行 python /Users/yupi/codex/skills/hatch-pet/scripts/prepare_pet_run.py --pet-name "Yupi Pengu...
正在思考

```

然后拆分成多个子任务并行处理，生成各种动作的精灵图帧（比如待机、奔跑、跳跃、失败等等），最后拼成一张完整的像素动画精灵图。

```

Use hatch-pet skill

The penguin's left/right identity is simple enough for mirroring: the motion row passed as consistent, and a framewise mirror gives the leftward drag without inventing a second gait. I'm deriving running-left with the skill script so timing stays in order.

已运行 python /Users/yupi/codex/skills/hatch-pet/scripts/deriv...
已关闭 1 个智能体

已运行 行 -r "[jobs[] | select(.status != "complete") | ],id, group...
已关闭 3 个智能体

The leftward row is derived and recorded. I'm launching the next two generated rows, waving and jumping, keeping the generation lane narrow so each worker can check its own strip carefully.

正在生成 2 个智能体
已根据以下指令创建 Hooke (worker):
Generate one hatch-pet row. Run dir: /Users/yupi/Documents/nameca...
已根据以下指令创建 Einstein (worker):
Generate one hatch-pet row. Run dir: /Users/yupi/Documents/nameca...

正在思考

```



等待很长很长一段时间之后，AI 终于完成了任务，然后就能使用自己制作的宠物啦



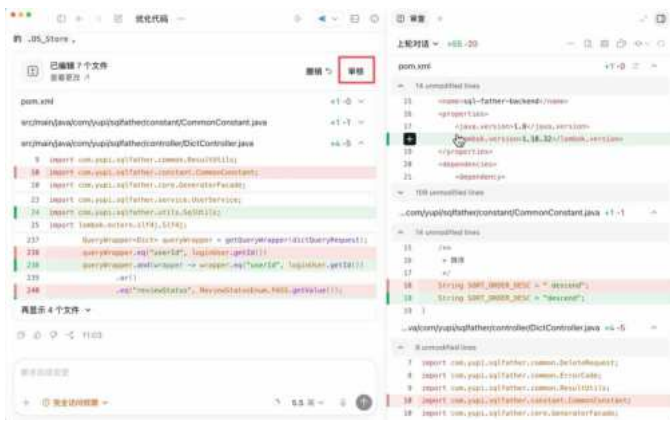
你还可以上传宠物到平台，和其他小伙伴分享。以后用 Codex 的时候，记得把我带在身边，保佑你 Bug 减减~



4.10 代码审查和版本管理

每次 AI 修改了文件，你都可以在侧边栏的「审核」面板里看到它改了什么。

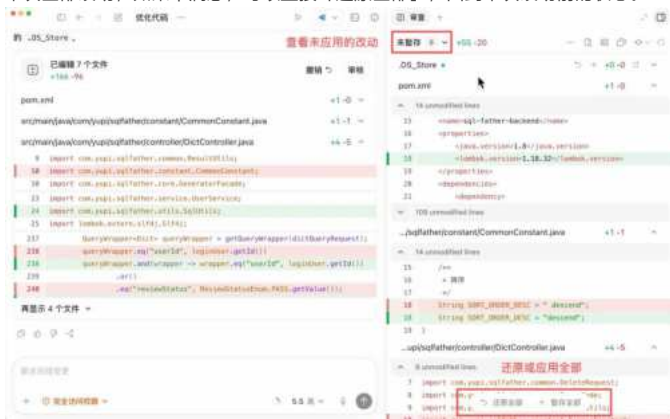
这个面板会列出所有被改动的文件，展开能看到具体增加了哪些内容、删除了哪些内容。



应用和撤销修改

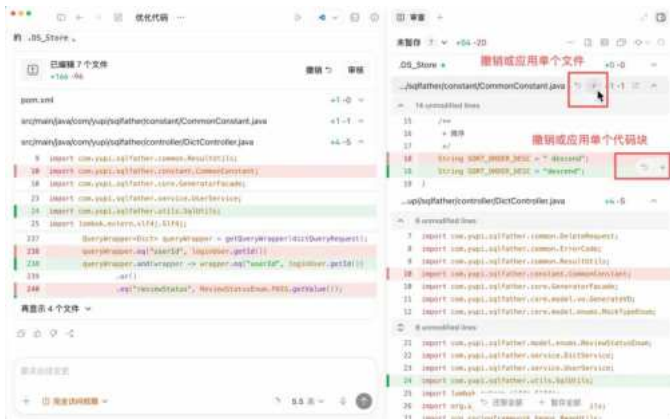
如果你想决定哪些代码要保留，可以选择查看「未暂存」的文件，在这里你可以灵活应用和撤销修改。

大多数情况下，你不需要自己看代码，直接点击「暂存全部」就行，相当于认可了本次全部改动；如果不满意，可以直接「还原全部」，回到本次改动前的状态。



如果你不满意某个文件的改动，直接点旁边的「还原」按钮就能恢复原样；满意的话点「暂存」就能标记为待提交。

你也可以只保留部分改动。每个文件的改动会被自动拆分成多个代码块，每个代码块旁边都有独立的「暂存」和「还原」按钮，你可以逐块决定哪些保留、哪些丢弃。



确定好哪些代码要暂存之后，可以「提交」暂存的改动。提交就相当于给代码存了个档，确认这次的修改是你想要的。

Codex 还内置了提交代码、推送到远程仓库、创建 Pull Request 的能力，不用离开 APP 就能完成整个代码管理流程，非常方便。



工作树

如果你是专业的开发者，可以试试「工作树」模式。你可以在创建新对话时选择启动模式为「新工作树」：

我们该构建什么？



这样 AI 会在一个隔离的分支中工作，不影响你当前的代码。很合适同时让多个 Agent 在同一个项目上并行干活，减少冲突。



GitHub 插件

如果你的项目托管在 GitHub 上，建议安装 GitHub 插件，可以直接在 Codex 里查看仓库信息、创建 Pull Request、做代码审查等等。

比如让我帮我查一下自己最受欢迎的开源项目：

@GitHub 帮我查看自己公开的 Star 数最多的前 10 个项目

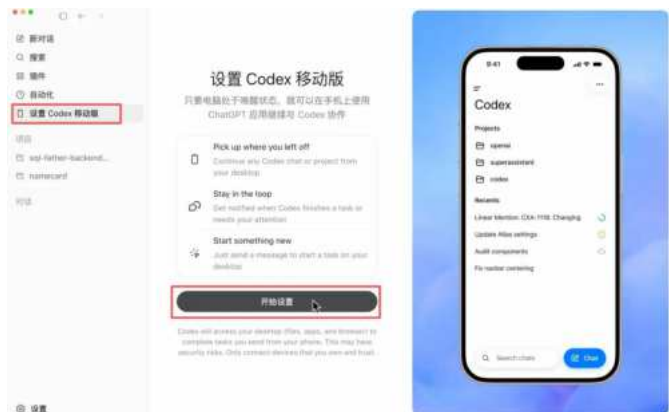
已处理 38s

我按 GitHub 插件返回的仓库 owner liyupi，并用 GitHub 实时公开数据按 Star 数降序重了前 10 个公开项目：

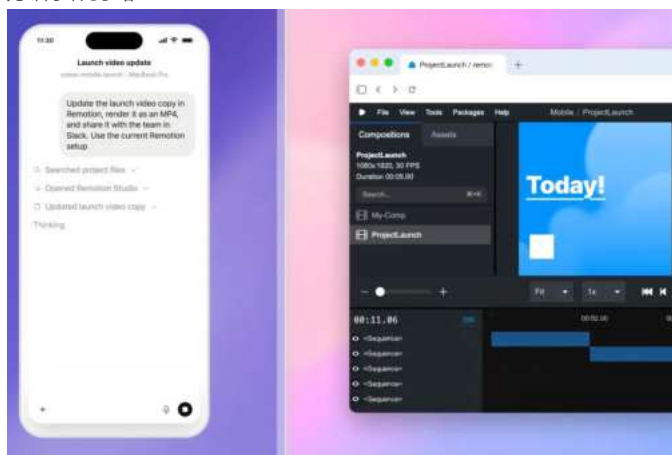
排名	项目	Stars
1	liyupi/ai-guide	14,236
2	liyupi/codefather	7,418
3	liyupi/mianshiya	5,451
4	liyupi/sqj-mother	4,164
5	liyupi/free-programming-resources	3,511
6	liyupi/sqj-generator	3,442
7	liyupi/code-nav	2,722
8	liyupi/yu-ai-agent	2,281
9	liyupi/yuindex	2,124
10	liyupi/sqj-father-backend-public	2,098

4.11 手机远程操控

Codex 最近新上线了一个很酷的功能，用手机控制电脑上的 Codex APP 干活。设置方法很简单，在电脑端 Codex 点击「设置 Codex 移动版」，然后点击开始设置，屏幕上会显示一个二维码。

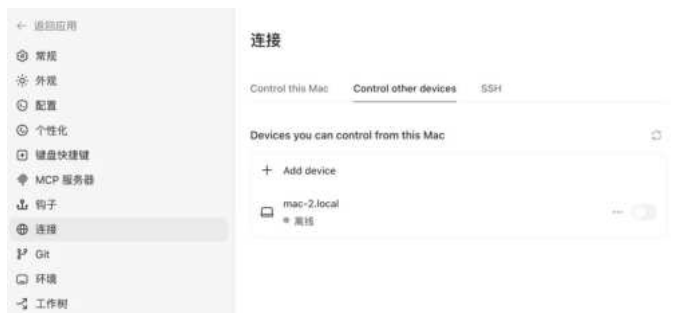


然后在手机上打开 ChatGPT App (确保更新到最新版)，扫描这个二维码，在 ChatGPT 中确认账号和工作区，完成可能需要的多因素认证步骤，就连接成功了。连上之后，你可以随时随地通过手机给电脑上的 Codex 下达任务、审批 AI 的操作请求、查看执行进度、检查生成的代码和结果。你的项目文件、已安装的插件和技能、所有配置都在那台电脑上，手机相当于一个远程遥控器。有点 OpenClaw 小龙虾内味儿了哈？



除了手机连接电脑，你还可以从另一台 Codex App 设备继续未完成的工作。比如你在公司用台式机跑了个长任务，回家后在笔记本上打开 Codex 就能接着看结果、继续操作。

你可以在 Codex 的连接设置中管理已连接的设备，比如断开某个手机的连接、设置是否保持电脑唤醒等等。



恭喜，看到这里你超过了 90% 的同学！

Codex 使用感受

学会了怎么用 Codex 之后，我也分享一下自己使用 Codex 的感受吧。

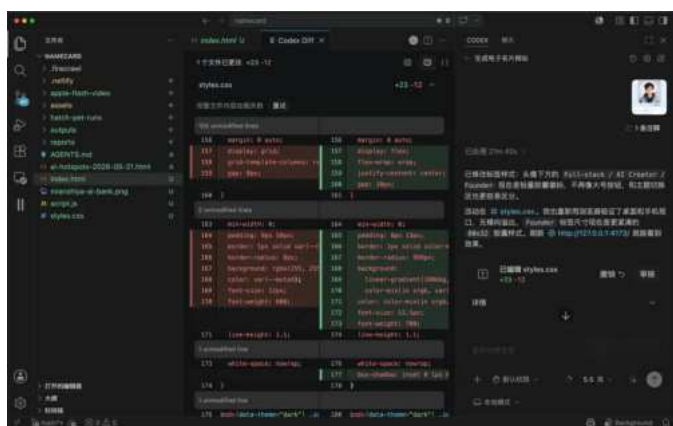
Codex 适合零基础或者不想折腾 IDE 和命令行的同学，下载就能用，使用门槛很低。尤其适合已经是 ChatGPT 付费用户的人，不用额外花钱就能拥有一个全能 AI 助理。

我个人非常推荐大家体验一下，尤其是非程序员，这可能是目前最适合你的 AI 编程工具。

但对我来说，目前 Codex 还不能完全满足我日常 AI 编程的所有需求。

作为一名程序员和 AI 编程博主，我需要阅读和编辑代码、精细控制某一行改动、随时切换不同的 AI 模型来对比效果，而这些 Codex 目前做得还不够好，没有内置代码编辑器、阅读代码的体验也一般。

所以实际使用的时候还是会搭配 VS Code 代码编辑器，或者很多任务直接在 Cursor 里搞定。



而且 Codex 的版本管理完全依赖 Git，项目必须是 Git 仓库才行，新手有一定门槛。

切换模型不够灵活，桌面端默认只有 OpenAI 自家模型，想用其他的得去配置文件或者用 CC Switch 之类的可视化工具。

而且目前对我来说最致命的是，GPT-5.5 在 Codex APP 的订阅模式下有效上下文只有 258K，做大型项目重构时跟其他 AI 编程工具和模型支持的 1M 上下文比，还是有差距。

不过每个工具都有自己的定位。Codex 更像一个什么都能做的「AI 助理 APP」，门槛最低；Cursor 更像一个兼顾代码编辑体验的「AI IDE」；Claude Code 更像终端里推理能力最强的 AI 大脑。

这几个工具的使用逻辑都差不多，学会一个再上手其他的会很快。

写在最后

OK 就分享到这里。

看完这期，你应该已经掌握了 Codex 从基础到进阶的所有核心用法，足够应对日常的编程和办公需求了。

其实 Codex 还有更多玩法和技巧，比如 Subagents 并行加速、Fork 分叉的妙用、自定义模型接入、Hooks 生命周期钩子这些。

如果本期视频点赞过万，我会尽快爆肝出 Codex 高级技巧篇！

对了，如果你想系统学习 AI 编程，可以看看我的免费开源教程《Vibe Coding 零基础入门教程》，上千张图、几十万字，从 0 开始带你学会 AI 编程。本篇教程的文字版也会收录其中。

开源指路：<https://github.com/liyupi/ai-guide>

为什么这么难?

因为不同的 AI 编程工具:

1. 能力差距太大: 有的工具只能生成简单代码, 有的能帮你做整个项目
2. 适用场景不同: 有的适合做课程, 有的适合做产品, 有的适合学习
3. 成本差别明显: 有的完全免费, 有的每月要几百块
4. 学习难度不同: 有的上手就能用, 有的需要一定基础

选对了工具, 你可以事半功倍; 选错了工具, 可能会让你觉得“Vibe Coding 也不过如此”。

二、AI 编程工具的 3 大类型

在深入了解具体工具之前, 我们先来看看 AI 编程工具的基本分类, 根据使用方式和复杂度, 我们把它们分为 3 大类:

零代码平台

在其网页上开箱即用, 不需要安装任何软件, 不需要懂任何代码, 适合完全零基础的新手, 能快速做出原型的产品。