

03期破局行动 | AI极速赋能编程 (含Cursor)

AI破局

内容出品人：熊猫Jay、雨飞、大圣、加加、小潘

手册出品方：AI 破局俱乐部团队

出品时间：2024 年 10 月 28 日

手册使用说明：本手册旨在向你展示一个项目的更多可能性，帮助你更好地理解或实操。

手册出品方介绍：AI 破局俱乐部是 AI 领域学习实践的头部社群，社群里不只有免费的专栏课程，还有许多与 AI 相关的赚钱玩法，实践案例，以及能够代表 AI 行业发展的最前沿讯息，能够帮助星友快速觉察到 AIGC 领域的最新赚钱机会。

知识星球

微信扫码加入星球 ▶



AI破局

距离结营还剩：

00 : 00 : 00 : 00
天 时 分 秒

学前须知

写在前面

欢迎大家来到 10 月破局行动营 | AI极速赋能编程 (含Cursor) | 实战手册，相信在接下来的日子里，大家都能学有所成。

本次破局行动营手册更适合：

- 1、程序员（本职工作提效、拓展）
- 2、产品人员（简单工具的制作1-3天都可完成）
- 3、自学过编程，并做出过一些小应用
- 4、0基础，但对编程非常感兴趣，并且有耐心有逻辑思维

实践出真知，希望大家都可以跟着手册一步步进行实操，从坚持日更开始，打开前路。本期开始时间-结束时间为**10月28日-11月19日**，采用的学习方式是**图文手册+教练直播分享+教练群内答疑+大作业**

为了鼓励大家实践，本次行动营增加了“**行动营大作业**”的活动，评为优秀作业的圈友将获得晶体奖励~

作业提交时间：11月11日-11月18日

作业提交入口：<https://d16rg8unadx.feishu.cn/share/base/form/shrcn7PjmaoXoGSTWBVYCqVkwac>

手册内需要复制的内容：[AI极速赋能编程 | 03破局行动 \(可复制版\)](#)

以下内容由 AI 破局俱乐部联合圈友制作而成，仅供行动营学员以及星球圈友学习使用。同时也欢迎圈友们在实践过程中持续反馈，和我们共同完善~

意见箱: <https://d16rg8unadx.feishu.cn/share/base/form/shrcnAGunfx2OjgHZ091sw8ipaf>

AI破局

作业要求

! 下面是本次行动营的作业安排, 请认真阅读~

◆ 常驻小尾巴【作业打卡】:

登录破局行动网站: <https://breakout.findyiai.com/action>

找到自己报名的项目, 按照打卡格式完成打卡。

打卡详情 [学习指南 | AI破局03期行动营](#):

1. 行动营21天需要打卡至少12次, 建议你在每周至少完成4次打卡, 保持持续行动的状态。
2. 每天1次打卡机会, 错过了不能补卡, 如未满足打卡12次, 保证金不退; 中途退出不退保证金。
3. 水日志为无效日志, 不计入正常的统计中, 水日志界定范围:
 - a. 完全乱写的驴唇不对马嘴, 或者全是标点符号或者无意义字符的;
 - b. 写和项目没关系的, 写人生哲理的;
 - c. 直接复制他人精选日志, 直接复制高手分享内容, 只复制没有任何想法和思考的内容
 - d. 多个项目共用一个作业内容的;
 - e. ...
4. 每周会有【高手分享】, 也会有问答和群聊精华汇总同步更新;
5. 在群内提问时, 建议具体描述你的问题, 并加上 #举手 标签, 方便圈友和教练更快定位到问题, 并做出解答。也欢迎你来帮助其他学员解决问题, 创造更大的价值。
6. 尊重行动营规则, 与他人建立有效链接, 不得恶意爆粉, 如被发现或被举报, 将抱出项目群。

分享时间	分享教练	分享主题	分享形式	直播回放链接
10.29 周二	大圣	AI时代的编程入门: 写给零基础的你	小鹅通直播	
11.3 周日	熊猫	cursor 快速上手与实战演练	小鹅通直播	
11.7 周四	雨飞	Cursor实战: 谷歌插件从入门到精通	小鹅通直播	
11.16 周六	加加	Cursor高手速成: 轻松实现复杂创意想法	小鹅通直播	

见破局系统【教练分享】部分

AI破局

★ 学习安排:

◆ **Part 1 AI编程入门:**

- 1) [AI时代的编程入门: 写给零基础的你](#)
- 2) [拥抱AI编程](#)
- 3) [准备工具Cursor, 并熟悉基础功能](#)

◆ **Part 2 AI编程实战:**

1) 用AI学编程解答常见疑问

加餐: Python 环境搭建

2) 开发小游戏

3) 开发谷歌插件/爬取资源

Part 3 AI编程进阶:

1) 接入大模型API

2) Coze 插件 + Cursor

3) Cursor + Devbox部署上线小产品

4) 建设网站

5) 🎉 大作业要求

AI破局

一、AI时代的编程入门：写给零基础的你

大家好，我是大圣，是本次破局行动的AI编程行动营教练

这次分享是基于我的一个开源的AI编程基础教程。

本次课程偏向于编程小白，听完这次课后，希望可以让你对AI编程有一个全面清晰的认知。

为接下来学习行动营的其他课程打下良好的基础

1.1 引言

"我这个编程小白，真的能学会AI编程吗？"

这可能是很多人心中的疑问。

在传统编程时代，学习编程确实不太容易 —— 要记忆各种语法规则，理解复杂的概念，敲很多代码...

但是现在不一样了。

AI正在逐渐降低编程的门槛，你不需要成为专业的程序员，也能用编程来实现自己的想法。

1.1.1 谁适合学习AI编程？

传统开发者

已经是程序员的朋友无疑是最适合学习AI编程的群体了

利用AI编程，你可以：

- 把重复性的编码工作交给Cursor等AI助手，自己专注在更有价值的设计工作上
- 用AI工具辅助进行性能分析，代码框架，避免写出屎山代码
- 在技术快速发展的今天，及早掌握AI编程也是保持竞争力的重要方式

产品经理

在我接触过的产品经理中。很多人都在为一些问题困扰：

- 想法很好，但总是很难快速验证
- 静态的原型稿没有办法帮助表达自己的产品框架
- 总是被技术的方案牵着鼻子走

学会AI编程后，你能做的事情会变得很多：

- 有了新想法，可以直接用Coze等平台快速做出一个简单demo，用实际效果来验证想法
- 在跟技术团队沟通时，能更好地理解技术可行性，设计出更实用的功能
- 产品原型不再局限于静态界面，可以直接用Cursor等AI工具生成可交互的演示版本

交互设计师

设计师朋友们应该都有这样的经历：辛苦设计的交互方案，在开发实现时总是差强人意。

学习AI编程能给设计工作带来新的可能：

- 可以自己动手做出简单的交互原型，让产品和技术同事直观体验设计理念
- 理解技术实现的过程，设计方案会更接地气
- 用动态演示替代静态原型，让每个人都能准确理解你的设计意图

宝爸宝妈

最近有位妈妈找我咨询，她的孩子才6岁，但对编程很感兴趣。问我需要学那种语言

在AI时代，孩子学习编程的方式也可以变得更有趣。

建议家长们可以这样做：

- 不用着急让孩子学习具体的编程语言，可以先用像Coze这样的平台培养编程思维
- 让孩子发挥想象力，逐步用AI工具把天马行空的想法变成现实
- 如果有时间，家长也可以一起学习，这将是陪伴孩子成长的绝佳方式

1.1.2 本次课程面向的对象

这门课程主要面向两类学习者：

PS：本节内容会偏向于编程新人

编程新人

如果你是编程新人，请记住：

- 🎓 这节课的目标不是把你变成程序员，而是教会你利用AI来实现想法
- 📄 课程会用最通俗的语言解释编程概念
- 🚶 这是一个循序渐进的过程，不要期待一蹴而就
- ⏳ AI编程不同于AI绘画、写作，需要持续学习和积累

传统开发者

如果你已经是程序员：

- 帮你掌握AI辅助开发的技巧

1.1.3 学习AI编程的平台

在小白入门AI编程的时候，一上来就接触AI编程工具并不是一个很好的选择。

市面上的AI编程工具：例如国内字节的MarsCode或者国外的Cursor

它们的目的是帮助已经学会编程的同学提高开发效率，但对于对编程基础概念都还不是很清楚的同学，这些工具并不友好

我们将AI编程学习分为两个阶段，并给大家推荐下适合小白入门和进阶的学习平台和工具

第一阶段：新手期

特点：对于编程的基本概念还不清楚，例如：基础的Python语法、变量、数据类型以及JSON等

这个时候推荐使用字节的Coze智能体平台进行学习。

在搭建智能体工作流的过程中，你会使用插件节点、数据库节点、变量节点等，这些节点都蕴含着编程的概念，但是又不需要你编码。在实践的过程中就将这些概念掌握了

同时，当你进阶到一定程度，Coze中也会有代码节点让你体验一把写功能性代码的成就感

第二阶段：实战期

当你有了AI编程的基础，想使用代码开发一个完整的应用时，Cursor就可以大显身手了

当到了开发完整应用的阶段，你开始编写多个函数或者是多个代码文件。

这时候一款好用的AI编程工具可以大大提高你的开发效率

1.1.4 遇到问题怎么办

不论是小白还是资深开发者学习AI编程，一定会遇到很多问题

这时候记住一句话：**遇事不决问AI**

我们是AI编程，不是传统编程，要善于利用大模型的强大能力！

- 🤔 看不懂代码？让AI给你逐行解释
- ✖ 代码报错了？把错误信息给AI分析
- 📚 概念不理解？让AI用类比帮你解释

1.2 编程是什么

不管是传统编程，还是AI编程，我们都得了解编程是什么。

1.2.1 为什么会有编程

想象你在教一个特别听话的助手做事。但是这个助手有如下几个特点：

- 非常准确，但有点“死板”
- 只会按照指令行事
- 需要非常清晰的步骤说明

这个助手就是计算机，而编程就是告诉计算机具体该怎么做事的过程。

所以编程的核心目的就是：指挥计算机按照人类的意愿行事，解决人类的特定问题

1.2.2 编程语言

编程就是给计算机下达指令的过程。

那么，计算机该如何听懂我们的指令呢？这就引出了一个关键概念：**编程语言**。

编程语言就像人类语言一样，种类繁多。常见的有：

- Python
- Java
- C++
- JavaScript
- 等等

想象这样一个场景：你的领导说：“XXX，今晚加班把这个任务完成”，你说：“好咧”

这里：你的领导就是程序员，你就是计算机，你的领导通过一句话就让你乖乖的把活干了~

程序员给计算机下达指令的过程也是一样的。

你可能会问：“为什么会有这么多编程语言呢？”，因为每种语言都有其特定的应用场景和优缺点。

这里不过多介绍，如果你还没有接触过任何一门编程语言，建议你直接选择Python

1.2.3 IDE以及插件

IDE是专业软件开发工程师的必备武器，但是对于小白来讲可能会有些困惑

简单来讲：IDE就是程序员写代码的地方，就跟你用Word写文档一样

你为什么不用txt，而是用Word写文档，因为Word功能强大，可以加粗，斜体，区分正文和标题等等

同理，程序员也不用txt写代码，而是使用配套了各种方便程序员写代码的IDE工具。

IDE全称：“集成开发环境”

1. Integrated (集成)：意味着它将多种开发工具整合到一个应用程序中。
2. Development (开发)：表明它是为软件开发而设计的。
3. Environment (环境)：指的是它提供了一个完整的工作空间，用于编写、测试和部署软件。

这个全称概括了IDE的核心特点 - 它是一个将多种编程工具和功能集成在一起的软件开发环境，旨在提高程序员的工作效率和便利性。

而再强大的IDE都不会满足所有开发者的需求，于是便有了插件的诞生。

插件用来扩展或者增强IDE的功能。

就好比你的手机已经有了拍照的能力，但是你还是要下载一个美颜相机一样。

了解了编程是什么之后，我们正式进入编程基础概念的学习。

1.3 编程的基础概念

有了AI，为什么我们还需要学编程的基础知识呢？

我给你举个例子：

在日常工作中，大家可能不止一次心里暗骂过自己的领导：“什么玩意，什么都不懂，还瞎指挥”

如果你连编程基本的概念都不知道，却总想着去指挥AI帮你编程，那我相信AI可能也在心里骂你

所以了解编程的基本概念，是你和AI有效沟通的基础

另外即使是最先进的AI也可能犯错。

如果你对编程一无所知，你可能无法识别AI生成的代码中的错误或不合理之处。

基本的编程知识能让你成为一个更好的“把关人”，确保最终的结果符合你的预期。

PS：请注意，编程新人不要期望下面的内容可以完全听懂，这块是需要不断实践的。

我希望你应该专注听我对每个概念的理解，而不是里面的细节。细节部分需要在后面不断的实践过程中学习消化

1.3.1 编程语言

首先我们需要了解一门语言的基础语法，这里我推荐Python

PS：编程语言了解一种即可，各种语言都是相通的，而且以后就算有了更高级的语言，只会学习起来更简单

学习的过程中请千万注意：

不要陷入到学习中无法自拔，尤其是Python，花3个小时了解基础即可，其余的时间要以练带学。

1.3.2 变量：程序中的“容器”

在编程语言概念中，变量是非常重要的概念

🌰 变量在编程中充当数据传递的媒介。具体来说，变量的主要作用可以分为两个阶段：

1. **存储数据**：将数据存储在变量中，使数据有一个名字标识，可以随时访问和修改。
2. **引用数据**：通过变量名来引用或使用存储的数据，从而在不同的地方或不同的时间点进行数据传递和操作。

我用一个例子来解释变量的作用：想象你有一个带标签的储物盒，这就是“变量”：

1. 盒子可以存放东西（数据）
2. 标签上写着名字（变量名）
3. 你可以随时更换盒子里的东西（修改数据）

下面这段代码可以简单的展示下变量的内容

```
1 # 创建一个名叫"age"的储物盒，放入数字18
2 age = 18
3
4 # 创建一个名叫"name"的储物盒，放入文字"小明"
```

```

5 name = "小明"
6
7 # 可以随时查看盒子里的内容
8 print(f"{name}今年{age}岁") # 输出: 小明今年18岁
9
10 # 可以随时更换盒子里的内容
11 age = 19 # 把age盒子里的18换成19
12 print(f"{name}明年{age}岁") # 输出: 小明明年19岁

```

我们说变量是用来承载数据的，那么数据又有哪几种类型呢？

1.3.3 数据类型：不同的“物品类型”

就像现实生活中的物品有不同类型：

- 文字 (String) : "你好", "Python"
- 数字 (Number) : 整数(1, 2, 3)或小数(3.14)
- 真假 (Boolean) : 对/错, 是/否
- 列表 (List) : [苹果, 香蕉, 橙子]
- 字典 (Dictionary) : {姓名: 小明, 年龄: 25}

其中列表和字典是Python中的概念，他们还可以被称作：JSON

我这里称之为复杂的数据类型，后面相对难学的也是这部分内容

JSON可以长下面这个样子

```

1 {
2   "name": "大圣",
3   "age": 18,
4   "city": "杭州"
5 }

```

或者这个样子

```

1 [
2   {
3     "name": "大圣",
4     "age": 18,
5     "city": "杭州"
6   },
7   {
8     "name": "艾木",
9     "age": 16,
10    "city": "深圳"
11  }
12 ]

```

学习JSON就类似学习一门简单的语法，这里我们不展开

我想跟你分享下在学习JSON时需要记住的两点：

- JSON跟我们的电脑文件夹结构很像，是一个树状结构，可以一层层的嵌套
- JSON与编程语言无关，不同编程语言都认识JSON，就像英语是通用的国际语言

1.3.4 控制流程：程序的决策与重复

条件判断 (if-else)

就像生活中的决策：

你早上起床时查看天气：

- 如果下雨，则带伞
- 否则，则带太阳镜

用代码实现如下：

```
1 # 下雨决策
2 if is_raining:
3     take_umbrella()
4 else:
5     wear_sunglasses()
```

循环 (for/while)

重复做某件事情

for循环

想象你有几个朋友，想给他们发一个生日祝福。你可以使用循环来一次性给每个朋友发送消息。

就是微信中的群发能力

```
1 friends = ["小明", "小红", "小刚"]
2 for friend in friends:
3     print("祝", friend, "生日快乐!")
```

while循环

假设你想要每天都进行锻炼，直到你瘦了10斤。在这种情况下，使用while循环会更合适。

```
1 currentWeight = 100 # 当前的体重
2 goalWeight = 90 # 目标体重
3
4 while currentWeight > goalWeight:
```

```

5     print("今天锻炼! ")
6     currentWeight--;
7
8 print("恭喜! 你完成了锻炼")

```

1.3.5 函数

函数是编程中非常重要的一个概念。简单来说，函数就是一段可以重复使用的代码。

函数就好比日常生活中的功能模块。

假设你是会计，你每天都要做各种加减乘除。

为了方便，你购买了一个计算器，这个计算器就相当于一个函数：

1. 你输入数字（输入）
2. 计算器内部进行计算（处理过程）
3. 显示计算结果（输出）

来看一个简单的代码例子：

```

1 # 定义一个简单的计算器函数
2 def calculator(num1, num2, operation):
3     if operation == "+":
4         return num1 + num2
5     elif operation == "-":
6         return num1 - num2
7     elif operation == "*":
8         return num1 * num2
9     elif operation == "/":
10        return num1 / num2
11
12 # 使用这个计算器函数
13 result1 = calculator(10, 5, "+")
14 print(f"10 + 5 = {result1}") # 输出: 10 + 5 = 15
15
16 result2 = calculator(10, 2, "*")
17 print(f"10 * 2 = {result2}") # 输出: 10 * 2 = 20

```

函数的主要作用：

1. **代码复用**：把经常要用到的代码打包成函数，避免重复写同样的代码
2. **模块化**：把复杂的问题拆分成小函数，让代码更容易理解和维护
3. **封装细节**：使用函数的人不需要知道内部具体是怎么实现的，只需要知道怎么用

在实际开发中，一个程序往往会包含很多个函数，每个函数负责处理特定的任务。

就像你的办公桌上可能除了计算器，还有打印机、扫描仪等不同的工具，每个工具都专注于完成特定的工作。

记住：理解函数并不需要你立刻就能写出完美的函数。

关键是要明白函数的基本概念 - 它就是一个能重复使用的功能模块，给定输入，经过处理后，产生特定的输出。

1.4 编程的思维模型

当学完了变量、数据类型、控制流程和函数这些基础概念后，让我们来聊聊编程的核心思维模型。

1.4.1 输入-处理-输出框架

在编程中有一个朴实但非常实用的框架：**输入、处理和输出**。

这个框架跟传统编程还是AI编程都没有关系，它是一个共性的东西。

理解了这个框架，你会发现阅读代码还挺简单的

我们用做菜来举个例子：

1. **输入**: 准备的食材
2. **处理**: 具体的烹饪步骤
3. **输出**: 最终的菜品

用代码来表达这个过程：

```

1 def cook_dish(ingredients):      # 输入: 食材
2     # 处理: 烹饪步骤
3     if not check_ingredients(ingredients):    # 检查食材是否齐全
4         return "食材不足"
5
6     prepared = prepare_ingredients(ingredients)    # 处理食材
7     cooked = cooking_process(prepared)            # 烹饪过程
8
9     return cooked       # 输出: 完成的菜品

```

这个框架为什么重要，因为到后面你会发现你在用AI帮你代码的时候，你要清晰的定义好输入和输出，而AI着重帮你搞定的则是处理的逻辑。

1.4.2 模块化拆分

你可能会问：“一个像豆包这样复杂的软件，也遵循这个框架吗？”

是的，一个有经验的程序员，会将复杂的功能拆解成多个模块。

而每个模块都遵循着输入-处理-输出的框架。

假设我们写一篇公众号，虽然流程有些复杂，但是可以拆分成如下几个模块

1. 选题模块

- a. 输入：当下相关领域的热点新闻
- b. 处理：根据某种规则进行处理
- c. 输出：本次的选题

2. 大纲确定模块

- a. 输入：选题
- b. 处理：AI
- c. 输出：文章大纲

3. 内容生成模块

- a. 输入：文章大纲和选题
- b. 处理：AI
- c. 输出：每一章的内容

4. 摘要生成模块

- a. 输入：文章内容
- b. 处理：AI
- c. 输出：文章摘要

5. 封面生成模块

- a. 输入：文章摘要 & 选题
- b. 处理：AI
- c. 输出：封面图片

1.4.3 流水线组装

当确定了每个模块（函数）之后，我们就可以像流水线一样将这些模块串起来组成一个功能

什么是流水线：

1. 第一个工作台（函数A）处理原材料，完成后把半成品放在周转箱（变量）里
2. 下一个工作台（函数B）拿到周转箱里的半成品，根据产品类型（if条件）选择不同的加工方式，有时需要重复加工几次（循环）
3. 就这样一个工作台接着一个工作台，最终制造出成品

我们用AI写公众号文章的例子来理解这个流程：

```

1 def generate_article(hotNews):
2
3     # 第一个函数：选题
4     topic = generate_topic(hotNews)           # 输入热点新闻，输出选题
5
6     # 第二个函数：生成大纲
7     outline = generate_outline(topic)         # 输入topic，输出outline
8
9     # 第三个函数：根据大纲生成内容
10    content = generate_content(outline)      # 输入outline，输出content

```

```

11
12     # 第四个函数: 生成摘要
13     summary = generate_summary(content)          # 输入content, 输出summary
14
15     # 第五个函数: 生成封面
16     post = generate_post(summary)                # 输入summary, 输出post
17
18     # 最终的文章包含这些部分
19     article = {
20         "topic": topic,
21         "outline": outline,
22         "content": content,
23         "summary": summary,
24         "post": post
25     }
26
27     return article

```

1.4.4 模块复用

当你理解了上面的概念，就会明白为什么程序员总是强调“模块复用”。

乐高积木就是典型的模块复用思想，通过一个个基础的小模块不同的组合，可以搭建出各式各样的作品。

Coze中的插件也是模块复用的一种体现

github上的一些开源库也是模块复用的体现

理解了这些思维模型，你再看那些复杂的代码，就会发现它们无非是这些基本概念的组合。

这时候，阅读代码就不再显得那么复杂和可怕了。

这些思维模型不仅适用于编程，在我们解决日常问题时也很有用。因为它们本质上是一种解决问题的方法论。

1.5 编程的进阶概念

了解了编程的基础概念和思维模型之后，我们就掌握了阅读代码的基础。

接下来我们需要了解三个跟代码不直接相关，但是在编程侧很重要的3个概念

1.5.1 跟外部程序交互：API接口

在AI编程中，有一个非常重要的场景，就是获取外部的数据。例如：

- 需要实时获取当前的天气
- 想在代码中引入豆包大模型的能力
-

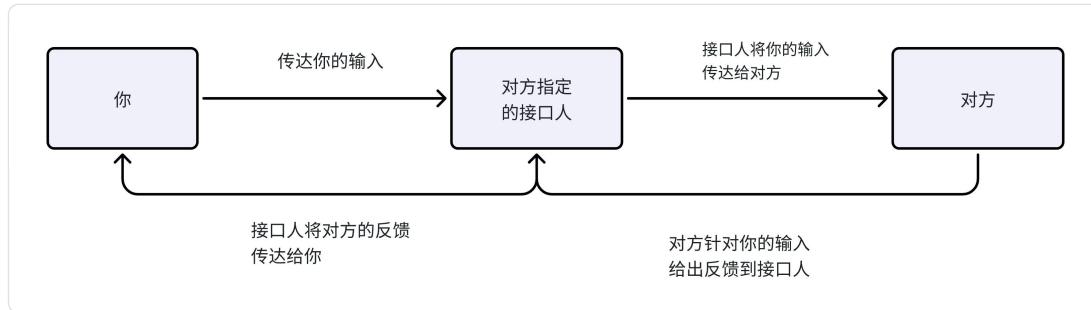
这时候我们就需要学习一个叫做API的概念，

API全称：应用程序接口

我们先来理解下什么是接口，我换个方式来说：接口人

当你因为工作需要和一个人谈业务的时候，但是对方不想直接跟你见面，所以对方指定一个接口人来传递消息

然后你所有信息的传达都是通过这个接口人来完成的



接口人是用来进行人和人之间传递信息的

那么API（应用程序接口）就是用来和应用程序进行传递信息的

那么应用程序又是什么呢？

- 豆包就是个应用程序
- 你用的各种软件都是应用程序

谁会和应用程序进行通信呢？

- 人可以和应用程序通信
- 其他应用程序也可以和应用程序通信

什么情况下需要和应用通信通信呢？

当你需要从某个应用程序获取数据的时候，我举个例子：

你搭建了一个网站，你想要实时显示当前的天气，那这个天气信息是不是得从气象局来。

于是你发现气象局对外提供了一个API，你每次调用都可以获取当前最新的天气信息。

只要你的网站接入了这个API，你就可以随心所欲的给你的用户展示天气信息了。

为什么在编程领域，API是一个非常重要的概念呢？

我们应该知道ChatGPT刚火的时候，国内也出现了好多的平台号称可以付费使用和ChatGPT的能力，这个就是我们说的套壳软件，这些平台肯定没有自己的大模型，他们只是接入了ChatGPT的API接口而已，甚至接入的都不一定是ChatGPT的能力

还有各种你在微信上见到的各种小程序，大多数的底层都是接入了一些开源或者付费的API，然后进行简单的前端页面包装就成了一个新产品了

所以如果你对API有一定的认知和了解，理论上来说你就可以通过像搭积木一样配合各种API组装出你的产品。现在好多的创业公司也是这么做的

1.5.2 数据存储

在一个正式运行的程序中，基本都离不开数据的存储

例如用户的账号密码、个人信息等都是需要进行数据存储的

这个时候，你需要了解一个叫做数据库的概念。

数据库就像是一个电子化的档案室，你可以：

- 安全地保管数据（比如用户密码会被加密存储）
- 方便地查找数据（像在Excel里快速找到某个信息）
- 随时更新数据（比如修改个人资料）
- 保证数据的可靠性（不会轻易丢失或损坏）

PS：新人接触数据库的时候就可以类比Excel，或者飞书的多维表格

1.5.3 应用部署

我们写代码的目的基本都是为了构建一个具有某个特定功能的程序，可能是APP，也可能是一个网站。

而将你的代码变成可以运行的应用程序，这个环节就叫做部署

当我们在电脑上写完代码后，这些代码目前只能在你自己的电脑上运行。

如果想让其他人也能使用你开发的程序，你需要把代码部署到服务器上。

服务器，简单来说就是一台24小时运行的计算机。

最常见的就是云服务器，比如阿里云、腾讯云提供的服务器。它们就像是租用的一台远程电脑，你可以把你的代码放到上面运行。

比如：一直很火的微信自动回复机器人，就是将代码部署在一台云服务器上，然后24小时提供服务。

部署的一般流程：

1. 选择一个云服务器
2. 在服务器上安装需要的环境（比如Python）
3. 把代码上传到服务器
4. 让代码在服务器上运行起来

PS：对于非程序员来讲，部署最好有手把手的教程，否则你会很痛苦....

🎁 【学习安排】跳转入口: ⭐ 学习安排

AI破局

二、拥抱AI编程 @大圣

上面我们花费大篇幅讲解了编程的基础概念，目的是帮助大家快速入门。

接下来我们终于进入到AI编程，我们首先必须对AI的特点有一个认知

- 它很强大，但需要清晰的指令
- 它很聪明，但需要正确的引导
- 它能写代码，但需要你说清楚要做什么

2.1 从执行者到方案设计

在AI编程时代，我们的角色发生了转变：

- 过去：我们是代码的"执行者"，专注于如何实现功能
- 现在：我们是方案的"设计师"，专注于想要实现什么功能

就像建筑师和建筑工人的区别：

- 建筑工人关注如何砌墙、怎么打地基
- 建筑师关注建筑的整体设计、功能分区、美观程度

我们不再需要掌握所有的编程细节，一些诸如语法之类的，随用随学即可。

但是我们要专注于：

- 理解编程的基本概念
- 明白代码实现的可能性
- **学会如何描述和拆解需求**

2.2 AI编程的核心技能

AI编程时代，最重要的技能不再是写代码，而是结构化思维 & 清晰的描述需求

结构化思考让你可以自顶向下思考，而清晰的描述需求则可以更好的指导AI干活。

两者缺一不可

结构化思维

还是以搭建积木为例：

- 不会一开始就漫无目的地拼
- 而是先看图纸，了解整体结构
- 然后按部件一步步组装

编程也是如此：

- 先设计整体架构
- 然后拆解成模块
- 最后逐个实现

清晰的需求描述

记住一个原则：**AI不是你肚子里的蛔虫，它需要具体的指示**

✖ 错误的需求描述：

"我要做一个类似淘宝的购物网站"

✓ 正确的需求描述：我想要做如下一个功能

```
1 1. 首页功能:  
2   - 商品展示区:  
3     * 每个商品卡片显示: 主图、标题、价格、销量  
4     * 支持按价格、销量、好评度排序  
5     * 提供分类筛选功能  
6  
7   - 搜索功能:  
8     * 支持按商品名称搜索  
9     * 支持按店铺名称搜索  
10    * 提供搜索历史记录  
11  
12 2. 商品详情页:  
13   - 图文展示区:  
14     * 支持多图片轮播  
15     * 显示商品详细描述  
16     * 展示规格参数表  
17  
18   - 交互功能:  
19     * 加入购物车按钮  
20     * 立即购买按钮  
21     * 收藏功能
```

2.3 与AI协作的正确姿势

记住这个公式：**好的输入 = 好的输出**

1. 先整体，后细节

- 先描述整个项目的目标
- 然后列出主要功能模块
- 最后深入每个模块的细节

2. 多用具体例子

- 告诉AI具体的使用场景
- 提供实际的数据样例
- 说明预期的展示效果

3. 善用反馈迭代

- AI的第一个方案可能不完美
- 及时提供具体的修改意见
- 通过多轮对话优化结果

2.4 AI编程工具

编程的目的不是为了写代码，而是为了能够创造出解决问题的应用。

写代码只是为了解决问题的一个手段而已。

这就像我们开车，目的是为了到达目的地，开车只是一个交通手段。

AI编程工具大致可以分为两类：

狭义的AI编程工具

这类工具主要面向程序员，专注于提升写代码的效率：

- Marscode、Cursor这样的AI编程IDE，就像是给程序员配备了一个AI助手
- 它们可以：
 - 帮你生成代码
 - 解释代码含义
 - 优化代码结构
 - 查找代码错误
 - 回答编程相关问题
- 这些工具让写代码变得更轻松

广义的AI编程工具

这类工具更进一步，它们让没有编程基础的人也能创造应用

- 像Coze、Dify这样的智能体搭建平台
- 它们提供了：
 - 可视化的界面搭建
 - 预设的功能模块

- 简单的逻辑配置
- 一键式部署功能
- 这些平台就像是提供了“积木”，让你通过组合这些积木就能搭建出实用的应用程序
- 它们极大地降低了创造应用的门槛

无论是哪种AI编程工具，它们都在服务同一个目标：让创造应用变得更简单。

想要创造一个简单的对话机器人，也许Coze就够用了。

需要开发一个复杂的企业应用？那你需要借助Marscode这样的专业工具。

核心始终是：专注于你要解决的问题，而不是沉迷于工具本身。

2.5 附录

产研团队的构成

在学习AI编程之前，先了解一个产品研发团队是怎么工作的，会对你的学习特别有帮助。

这就像你想拍一部短视频，了解编剧、导演、摄影、剪辑各自负责什么，能帮你更好地掌握整个创作过程

我用大家熟悉的美团外卖APP为例，聊聊研发团队的几个关键角色：

1. 老板说，我们要做一个外卖点餐APP，于是产品经理就开始规划

 **产品经理**就像总设计师，规划整个产品的功能，他们把用户的需求变成具体的功能清单。

- 用户能看什么餐厅、怎么筛选、
- 配送费怎么算
- 订单状态怎么展示。

2. 产品经理在设计出整体的产品框架之后，交互设计师就会登场

 **交互设计师**专注让产品好用：

- 点餐时菜品怎么展示最清晰
- 下单时哪些信息要重点显示
- 付款时优惠券放在哪里最显眼又不会误触

3. 当产品和交互就绪之后，流程就进入到开发手中，也就是我们常说的程序员

 常见的开发角色有两个：前端和后端开发

前端开发负责用户能看到的界面，就像餐厅的前厅：

- 让界面好看又好用
- 点餐界面要流畅不卡顿
- 优惠券要显眼但不突兀

后端开发处理看不见的核心功能，就像餐厅的厨房：

- 处理订单数据
- 计算配送费和优惠
- 保存用户信息

4. 当前后端开发完成后，流程流转到测试工程师手中

 **测试工程师**则像品控师，确保每个功能都正常运作

- 订单金额计算准确
- 支付流程不出错

如果你是**产品经理**，建议先了解后端：

- AI能帮我们生成界面设计，但产品的核心逻辑和数据架构更需要人来思考
- 比如美团外卖，AI可以设计界面，但“如何计算配送费”、“怎么匹配骑手”这些核心逻辑，需要你理解后端原理
- 了解后端还能帮你评估功能的开发难度，知道什么是真正可行的

如果你是**交互设计师**，建议先学前端：

- AI确实能生成界面，但好的交互体验需要理解前端实现原理
- 了解前端能让你设计出既美观又容易实现的方案
- 知道动画效果怎么实现，就能设计出流畅不卡顿的交互

如果是新人同学，没有特别喜好的情况下，建议从后端入手，先找找感觉

 【学习安排】跳转入口：  [学习安排](#)

 飞书文档 - 图片

三、AI编程工具——Cursor @熊猫

3.1 Cursor 与其他工具之间的区别

最近不少小伙伴都在问 Cursor 与 GitHub、通义灵码的区别是什么？

网上很多帖子讲得不是很清楚，所以我单独拿出一个章节来说明。

相比于其他编程工具，Cursor 的最大亮点是：**当你进入一个完整的项目代码中时，它可以索引所有代码的内容。**

这样，当你需要修改或新增任何代码时，它能快速了解背景，帮你得到符合预期的答案。

当然，你可以选择只引用部分文件，或者某些文件里的一段代码。这样，Cursor 就不会搜索整个代码库，避免无关的上下文干扰生成的结果。

比如，这样我选中了代码 `kamis.py` 的 30 ~ 42 行，这样 AI 就会用这段内容做为主要的上下文。

The screenshot shows a code editor with a Python file named 'kamis.py'. The file contains code for managing 'Kami' objects, including generating unique codes and saving them to a list. A 'CHAT' interface is open, showing a message from 'claude-3.5-sonnet' and a list of recent messages.

```
backend > kamis.py Current File x kamis.py (30-42) Code x
```

```
class KamiManager:
    def generate_kami_code(self, user, expiry_date, points):
        # 使用用户名、过期时间和积分生成唯一的卡密代码
        unique_string = f"{user}{expiry_date}{points}{uuid.uuid4()}"
        return hashlib.md5(unique_string.encode()).hexdigest()[:16]

    def add_kami(self, kami_data):
        new_kami = {
            'id': str(uuid.uuid4()),
            'code': self.generate_kami_code(kami_data['user'], kami_data['expiry_date'], kami_data['points']),
            'user': kami_data['user'],
            'expiry_date': kami_data['expiry_date'],
            'points': kami_data['points'],
            'status': 'unused',
            'created_at': datetime.now().isoformat()
        }
        self.kamis.append(new_kami)
        self.save_kamis()
        return new_kami

    def get_kami(self, kami_id):
        for kami in self.kamis:
            if kami['id'] == kami_id:
                return kami

    def save_kamis(self):
        with open('kamis.json', 'w') as f:
            json.dump(self.kamis, f)
```

Ask anything, ↑ to select

claude-3.5-sonnet @ Mention Image

no context chat codebase

All chats View all

- 保存用户密码的文... 1d ago
- 保存用户卡密的文... 1d ago
- CURSOR文件夹下... 1d ago
- 用户卡密信息保存... 1d ago
- 如何打开特定页面... 2d ago

如果之前小伙伴们使用过 ChatGPT、Claude3.5 进行编程，在这种情况下，除了要提交大量代码和输入背景信息作为上下文之外，还需要将生成的代码逐个塞进原来的文件里，这比较麻烦。

在 Cursor 中，基于需求生成的结果，点击一些按钮，就可以直接在对应的位置帮你修改好。

The screenshot shows a code editor with a Python file named 'kamis.py'. The code defines a class 'KamiManager' with methods for generating card codes, adding cards, and getting specific cards. A 'note' field is added to the 'new_kami' dictionary. Below the code editor is a terminal window showing a command-line interface with multiple 'zsh backend' sessions.

CHAT

在新增卡密的时候，帮我新增一个属性：备注，我可能需要记录一些零散的信息在备注字段中。

好的，我理解您想在新增卡密时添加一个备注字段。我们可以修改 `add_kami` 方法来实现这一点。以下是对代码的修改建议：

```
def add_kami(self, kami_data):
    new_kami = {
        'id': str(uuid.uuid4()),
        'code': self.generate_kami_code(kami_data['user'], kami_data['user']),
        'user': kami_data['user'],
        'expiry_date': kami_data['expiry_date'],
        'points': kami_data['points'],
        'status': 'unused',
        'created_at': datetime.now().isoformat()
    }
    self.kamis.append(new_kami)
    self.save_kamis()
    return new_kami

def get_kami(self, kami_id):
    for kami in self.kamis:
```

这里的修改主要是在 `new_kami` 字典中添加了一个 'note' 键。我们使用 `kami_data.get('note', '')` 来获取备注信息，如果 `kami_data` 中没有提供 'note' 键，则默认为空字符串。

这样，当您调用 `add_kami` 方法时，可以选择性地包含一个备注。例如：

```
kami_manager.add_kami({
    'user': '用户名',
    'expiry_date': '2023-12-31',
    'points': 100
})
```

此外，还有一个最重要的功能：**Composer**。

正常情况下，我们需要通过聊天去得到一个新功能的代码，然后创建文件，粘贴代码。

虽然一个一个确认让人更放心，但是功能复杂的情况下，涉及文件过多，就会变得很麻烦。

Composer 它可以帮助我们快速地、有序地创建多个文件，以满足我们的要求，效率将大大提升。



但是，这同样带来了风险。

当你的功能稳定后，要求 Composer 进行全局调整代码时，如果你没有仔细阅读就接受修改，一旦出错，将很难撤回。

当然，你可以选择继续要求 Composer 修改。Composer 目前还处于测试阶段，相信未来会更加好用。

所以，我的建议是：**最初写好用户故事，使用 Composer 创建好相应的代码结构，一旦功能相对稳定后，建议使用右侧的 Chat 去逐个迭代功能**，这样方便确定，稳定性更好。

不止于此哦~

当我们需要调整界面样式时，可以上传一张图片，让 Cursor 参考图片风格来调整代码。当然，从零开始创建也是可以的。

The screenshot shows the Feishu AI Composer interface. At the top, there's a 'CHAT' tab and a file tab labeled '+ index.html Current File'. Below the tabs is a preview area containing a colorful illustration of a cityscape with buildings and people, followed by a screenshot of a login page with a blue header and a QR code. To the right of the preview are 'Remove' and 'Collapse' buttons. A red box highlights the text '参考图片中的样式，帮我修改登录页面的风格。' (Refer to the picture style, help me modify the login page's style.) In the bottom left, there's a code editor window titled 'index.html' with the following code:

```
<!DOCTYPE html>
<html lang="zh">
<head>
    <!-- ... 保持原有的head内容不变 ... -->
</head>
<body>
    <div id="app">
        <div v-if="!isLoggedIn" class="login-container">
            <div class="login-content">
                <div class="login-left">
                    <!-- 这里可以添加一个类似参考图片的插图 -->
                    
                </div>
                <div class="login-right">
```

At the bottom right of the code editor are 'Ask', 'Copy', and 'Apply' buttons.

最后，它还有一个很好用的功能：AI 全局检索。

当处于一个大型代码库时，通过 Cursor 搜索代码会非常简单，它能告诉我们对应文件的位置以及与其他代码的关联性。

注意，使用时，点击 codebase，这样 Cursor 将从整个代码库中搜索。当我们不知道代码在哪里时，这个功能比传统工具会更加好用。



3.2 注册

进入 Cursor 官方：cursor.com，点击 Download for Free 进行下载。



点击下载后的安装文件，按照提示一步一步安装即可。



接下来我们需要注册账户，才可以使用 Cursor。

官方支持三种注册方式：**GitHub**、**谷歌账号**、**普通邮箱地址**。测试下来，发现也支持国内邮箱地址。



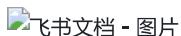
普通用户注册后，可以免费体验 14 天的 Pro 版本，拥有高级模型的 500 次对话机会，比如 claude3.5-sonnet、gpt-4o。

除此之外，可以无限使用 gpt-4o-mini 和 cursor-small 模型。



我们来完整了解下它的订阅模式：

- **免费用户**: 一共可以使用 2000 次普通模型的问答。
- **Pro 模式**: 20刀/月, 500 次高级模型问答, 不限次普通模型问答。
- **Business 模式**: 40刀/月, 全部不限制次数。



说实话，价格不低，但是有 14 天的免费体验时间，这给了我们充足的机会来判断是否需要付费。

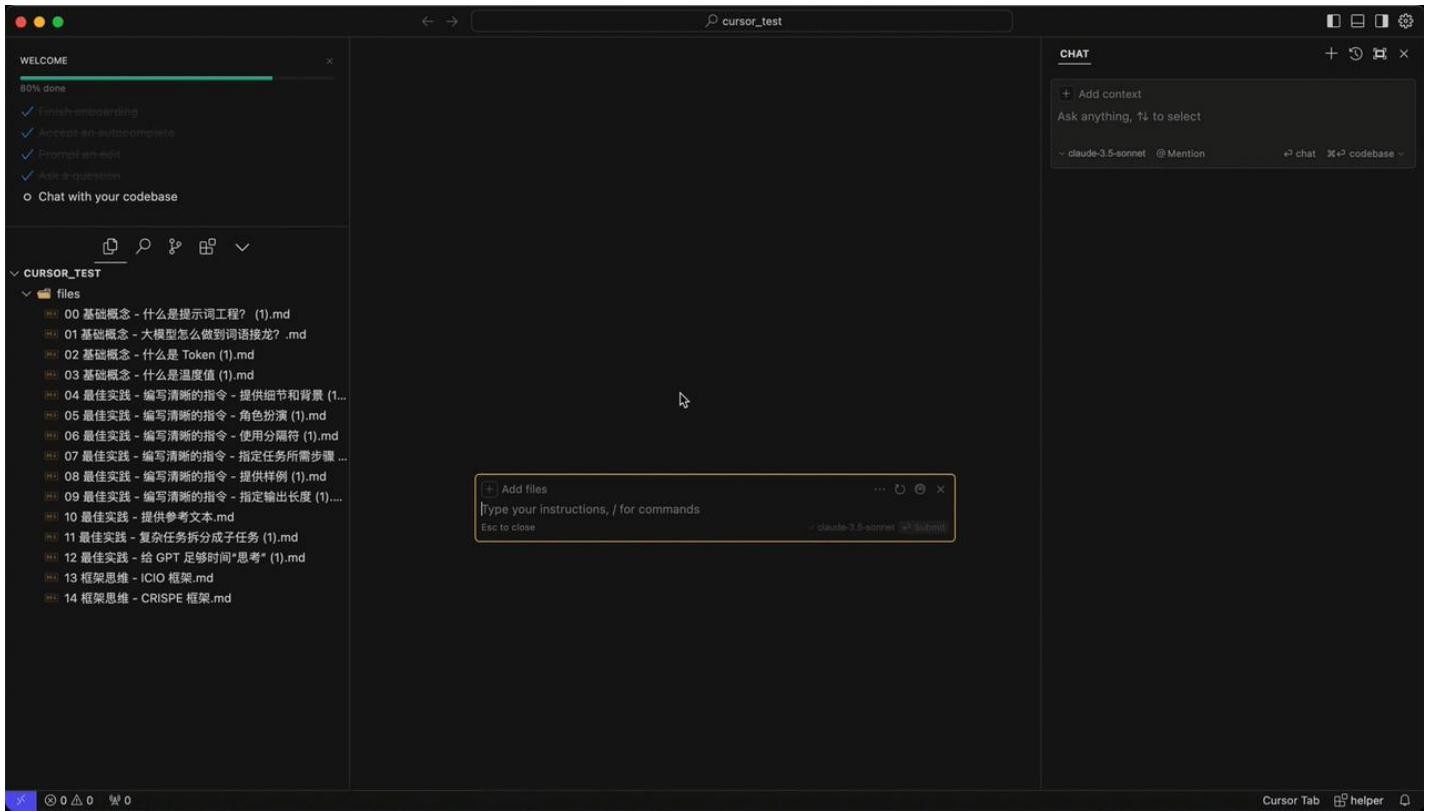
如果每个月能为我们创造远超过 20 刀的价值，这个付费就值得 ~

大家见仁见智吧。

3.3 使用

1、全局创建代码结构

Mac 用户使用 Command + i 唤醒 Composer，输入提示词后开始创建代码结构。



用浏览器打开 index.html 文件，我们看下最初的效果。



关于其他系统的快捷键，可以从 Cursor > Setting > Keyboard Shortcuts 快捷指令清单中查看。



2. 新增 / 修改代码、文字

选中代码，使用 **Command + K** 打开窗口，并输入修改的要求。

如果不选中代码，打开窗口，可以要求 AI 帮助我们实现新的功能。

这里我让 AI 帮我增加一个广告位。

The screenshot shows the Feishu Cloud Document interface. On the left, there's a sidebar with a 'WELCOME' section showing '60% done' and a list of checked items: 'Finish onboarding', 'Accept an autocomlete', 'Prompt an edit', and 'Ask a question'. Below it is a 'CURSOR_TEST' section with a 'files' folder containing several .md files, including '00 基础概念 - 什么是提示词工程? (1).md' through '13 框架思维 - ICIO 框架.md', and an 'index.html' file. The main area is a code editor for 'index.html' with the following content:

```

<html lang="zh-CN">
<head>
</head>
<body>
  <div class="container">
    <div class="file-list">
      <h2>文件列表</h2>
      <ul id="fileList"></ul>
    </div>
    <div class="file-content">
      <h2>文件内容</h2>
      <div id="fileContent"></div>
    </div>
  </div>
<script>
  // 模拟文件列表数据
  const files = ['文件1.txt', '文件2.txt', '文件3.txt'];

  // 显示文件列表
  const fileList = document.getElementById('fileList');
  files.forEach(file => {
    const li = document.createElement('li');
    li.textContent = file;
    li.onclick = () => loadFileContent(file);
    fileList.appendChild(li);
  });

  // 加载文件内容
  function loadFileContent(fileName) {
    const fileContent = document.getElementById('fileContent');
    // 这里应该是从服务器获取文件内容的代码
    // 为了演示，我们使用模拟数据
    fileContent.innerHTML = `<h3>${fileName}</h3><p>这是 ${fileName} 的内容。实际应用中，这`;
  }
</script>
</body>
</html>

```

On the right, there's a 'CHAT' panel with a message from 'claude-3-5-somnet' and a mention of 'codebase'. The status bar at the bottom indicates 'Ln 50, Col 5' and other document details.

This screenshot shows a simplified user interface for file management. It has two main sections: '文件列表' (File List) on the left and '文件内容' (File Content) on the right. In the '文件列表' section, there are three buttons labeled '文件1.txt', '文件2.txt', and '文件3.txt'. In the '文件内容' section, there is placeholder text: '这里是广告内容。您可以在此处放置您的广告。' (This is advertisement content. You can place your advertisement here.).

当然，除了代码，我们也可以选中文字进行修改，比如改写、翻译等等。

The screenshot shows the Feishu Cloud Document interface. On the left, there's a sidebar with sections like 'WELCOME' and 'CURSOR_TEST'. The main area is a code editor with the following content:

```

14 框架思维 - CRISPE 框架.md •
files > 14 框架思维 - CRISPE 框架.md > # 14 框架思维 - CRISPE 框架 > ## 四、总结
8   # 14 框架思维 - CRISPE 框架
49   ## 三、实践
58
59   1、初始化提示词
60
61   
62
63   2、提供数学题并得到两套答案。
64
65   
66
67   得到了两套相对通俗的数学题解法。效果基本满意 ~
68
69   ## 四、总结
70

```

A modal window titled '帮我翻译成英文' (Help me translate into English) is open over the code editor, containing the following text:

本章我们借助简单的数学题辅导案例说明了框架的使用方式，**CRISPE框架** 的灵活性使其可以处理相对复杂的任务，帮助用户获得多样化的方案和建议。
In this chapter, we have demonstrated the usage of the framework through a simple case of tutoring math problems. The flexibility of the **CRISPE framework** allows it to handle relatively complex tasks, helping users obtain diverse solutions and suggestions.

除此之外，还有很多其他的框架，大家可以自行搜索。
In addition to this, there are many other frameworks that you can search for on your own.

3、自动补全代码、注释、文字

输入代码或者注释，Cursor 会自动补全代码，按 Tab 将会生效。

The screenshot shows the Feishu Cloud Document interface. On the left, there's a sidebar with sections like 'WELCOME' and 'CURSOR_TEST'. The main area is a code editor with the following content:

```

index.html < html lang="zh-CN" >
  <body>
    <script>
      const fileList = document.getElementById('fileList');
      files.forEach(file => {
        const li = document.createElement('li');
        li.textContent = file;
        li.onclick = () => loadFileContent(file);
        fileList.appendChild(li);
      });

      // 加载文件内容
      function loadFileContent(fileName) {
        const fileContent = document.getElementById('fileContent');
        // 这里应该是从服务器获取文件内容的代码
        // 为了演示，我们使用模拟数据
        fileContent.innerHTML = `<h3>${fileName}</h3><p>这是 ${fileName} 的内容。实际应用中，这

```

The right side of the interface has a 'CHAT' sidebar with a message input field and a list of recent messages.

除了补全代码之外，还能补全文字，大家可以尝试一下。

2、提供数学题并得到两套答案。

```

```

得到了两套相对通俗的数学题解法。效果基本满意 ~

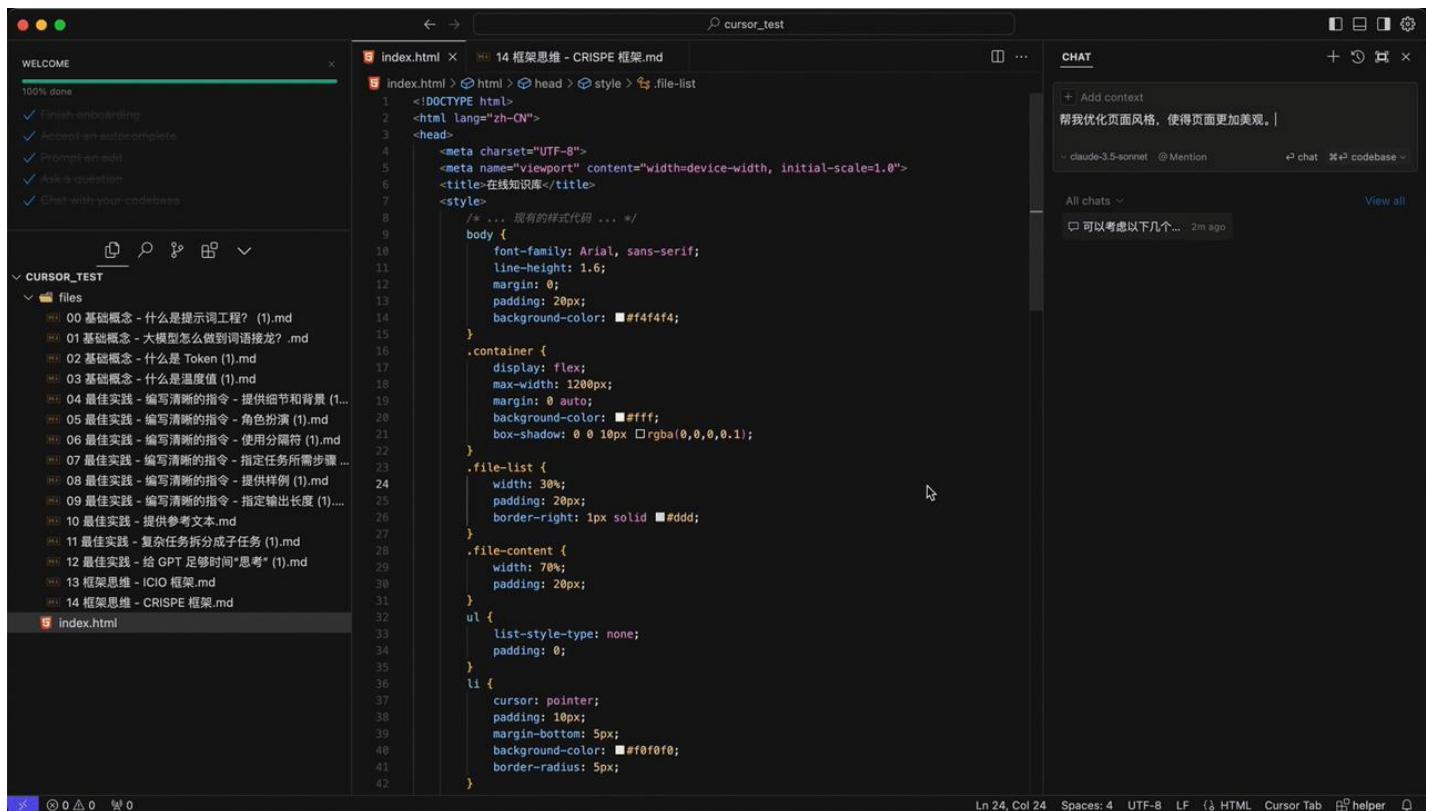
四、总结

本章我们借助简单的数学题辅导案例说明了框架的使用方式，**CRISPE框架** 的灵活性使其可以处理相对复杂的任务，帮助用户获得多样化的方案和建议。

除此之外，还有很多其他的框架，大家可以自行搜索学习。

4、对话窗口

Mac 使用 **Shift + Command + L** 打开聊天窗口，输入优化页面的需求，AI 可以为我们提供不同方案。



这里我倾向于使用好看的配色方案。点击 Apply，再点击 Accept，就生效啦。

对了，要记得保存文件后，否则刷新网页不会生效，Mac 的快捷键的 Command + S。

```

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   <title>在线知识库</title>
7   <style>
8     :root {
9       --primary-color: #3498db;
10      --secondary-color: #2c3e50;
11      --background-color: #ecf0f1;
12      --text-color: #34495e;
13      --hover-color: #2980b9;
14      --primary-color: #4a90e2;
15      --secondary-color: #34495e;
16      --background-color: #f5f7fa;
17      --text-color: #333333;
18      --hover-color: #3498db;
19      --accent-color: #e74c3c;
20    }
21
22    body {
23      font-family: 'Helvetica Neue', Arial, sans-serif;
24      font-family: 'Roboto', 'Helvetica Neue', Arial, sans-serif;
25      line-height: 1.6;
26      margin: 0;
27      padding: 20px;
28      background-color: var(--background-color);
29      color: var(--text-color);
30    }
31
32    .container {
33      display: flex;
34      max-width: 1200px;
35      margin: 0 auto;
36      background-color: #fff;
37      box-shadow: 0 0 20px rgba(0,0,0,0.1);
38      border-radius: 8px;
39      border-radius: 12px;
40      overflow: hidden;
41    }

```

当然，这不是一个成品，如果是想要做一个完整功能，我们需要不停的和 Cursor 进行对话。

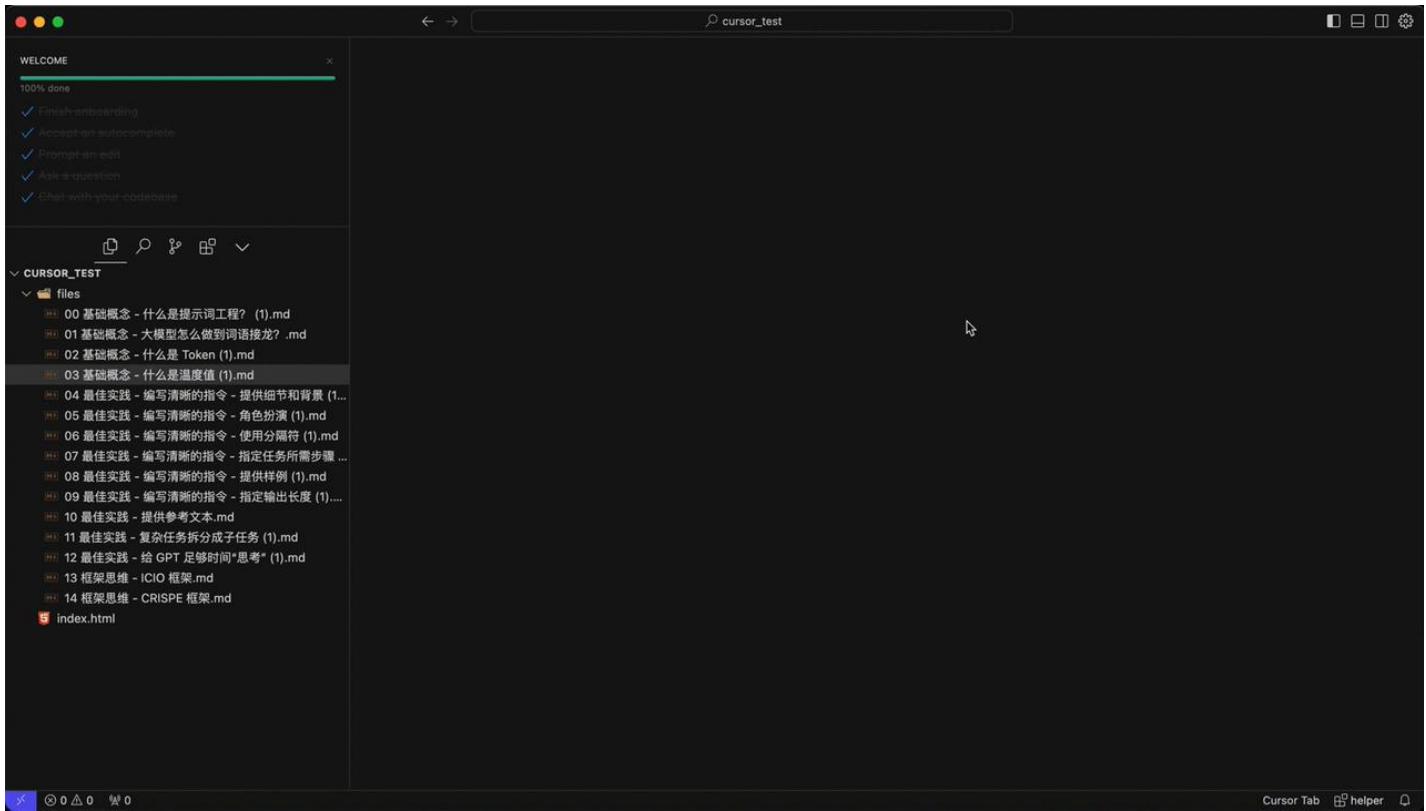
不过，别急，在案例部分，我会介绍完整的制作过程。

5、全局搜索

我们还可以把它当做一個简易的 AI 搜索工具，让它根据现有文件夹下的内容回答我们的问题。

比如这里，我问到基于文件内容，告诉我温度值设置的误区在哪里，是什么？

它的回答准确度很高，甚至能定位到具体文件的行。



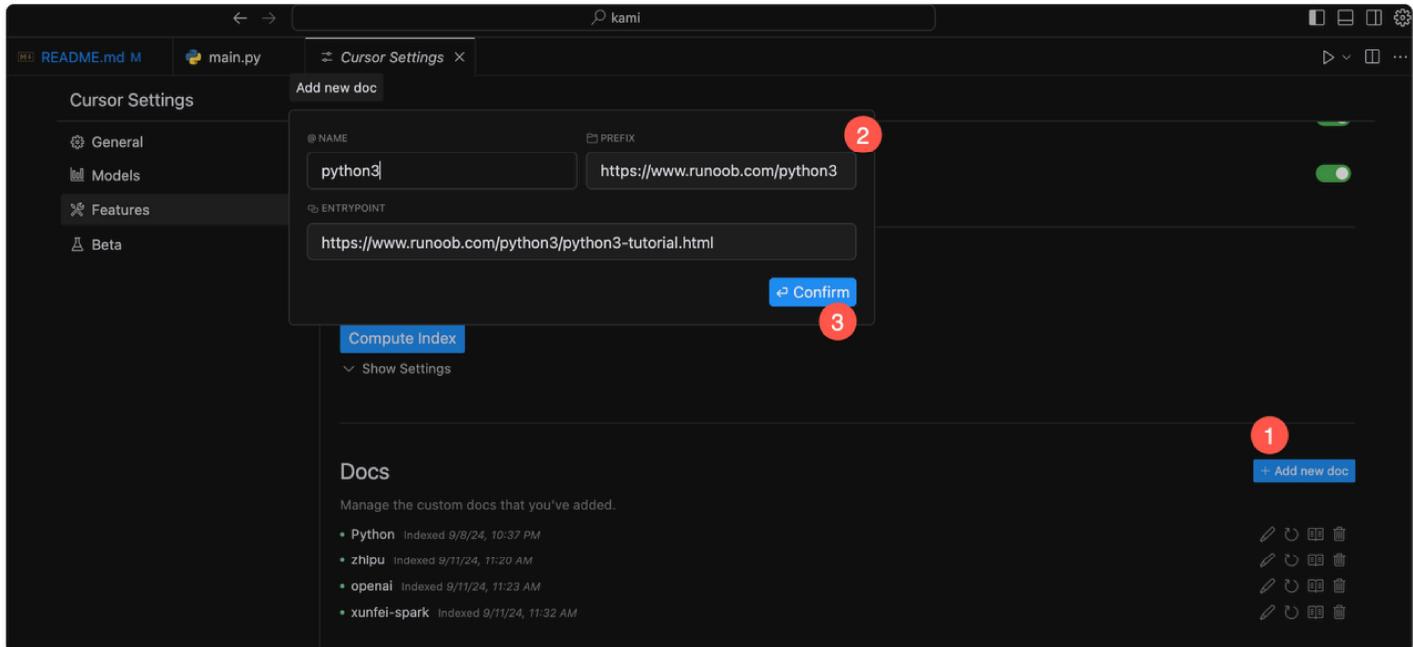
6. 自动爬取指定网页内容辅助开发

这个功能相对比较隐蔽。可以帮助我们爬取对应网页的内容，作为指南辅助我们进行开发，但是存在一些缺陷。

! 注意：静态页面爬取的成功率更高，非静态页面的话，可能爬取内容会遗漏，或者完全爬取不了。

操作方式：进入 **Setting > Cursor Settings > Features**，在 Docs 位置进行配置。

- Name: 指令名称
- Prefix: 待爬取网页地址的前缀，这样可以爬取多个网页内容
- EntryPoint: 网页地址



拿菜鸟教程为例，我们按照下图配置即可。如果一切正常，可以将所有 python3 作为前缀的网页内容爬取下来。

! 目前爬取多个网页内容的效果不是很稳定，单页面成功的概率更高。

The screenshot shows a user interface for managing custom documents. At the top, there are input fields for 'NAME' (python3), 'PREFIX' (https://www.runoob.com/python3), and 'ENTRYPOINT' (https://www.runoob.com/python3/python3-tutorial.html). A 'Confirm' button is visible. Below the input fields, a browser window displays the URL https://www.runoob.com/python3/python3-tutorial.html. To the right of the browser, a sidebar titled 'Python 3 教程' lists various Python 3 topics. The main content area is titled '查看 Python 版本' and contains instructions for checking Python version via command line (python -V or python --version) and an interactive session showing Python 3.3.2. A green arrow points from the browser's URL bar to the 'ENTRYPOINT' field.

点击 Confirm 之后，待爬取网页内容的数量越多，爬取的速度会越慢，大家耐心等待一会。

The screenshot shows the 'Docs' section of the application. It lists indexed items: Python, zhipu, openai, xunfei-spark, and python3. The python3 item is highlighted with a red box and has a tooltip 'Indexing Python3 注释 / 菜鸟教程...'. To the right of the list are icons for edit, refresh, view, and delete. A blue button '+ Add new doc' is at the top right.

爬取成功的状态是这样。

Docs

[+ Add new doc](#)

Manage the custom docs that you've added.

- Python Indexed 9/11/24, 12:24 PM
- zhipu Indexed 9/11/24, 11:20 AM
- openai Indexed 9/11/24, 11:23 AM
- xunfei-spark Indexed 9/11/24, 11:32 AM
- python3 Indexed 9/11/24, 12:23 PM



点击 **书本** 的图标，可以看见爬取成功的页面数量和名称。

Docs

[+ Add new doc](#)

Manage the custom docs that you've added.

- Python Indexed 9/11/24, 12:24 PM
- zhipu Indexed 9/11/24, 11:20 AM
- openai Indexed 9/11/24, 11:23 AM
- xunfei-spark Indexed 9/11/24, 11:32 AM
- python3 Indexed 9/11/24, 12:23 PM



Chat

Always search the web

This would be the same as having @web in every prompt

[End chat stream](#)

Indexed 16 pages

- Python3 教程 | 菜鸟教程
- Python3 字典 | 菜鸟教程
- Python3 集合 | 菜鸟教程
- Python VScode 配置 | 菜鸟教程
- Python3 简介 | 菜鸟教程
- Python3 解释器 | 菜鸟教程



我多轮测试下来，静态页面：比如菜鸟教程，爬取的成功率高，有几率将 100 多个网页都爬取下来。非静态页面：比如 **OpenAI 开发指南** 只能爬取一页，**智谱开发指南** 始终爬不全。

Docs

Manage the custom docs that you've added.

- Python Indexed 9/11/24, 12:24 PM
- zhipu Indexed 9/11/24, 11:20 AM
- openai Indexed 9/11/24, 11:23 AM
- xunfei-spark Indexed 9/11/24, 11:32 AM
- python3 Indexed 9/11/24, 12:23 PM

Indexed 147 pages

- Python ord() 函数 | 菜鸟教程
- Python3 shuffle() 函数 | 菜鸟教程
- Python3 rstrip() 方法 | 菜鸟教程
- Python3 ceil() 函数 | 菜鸟教程
- Python3 多线程 | 菜鸟教程
- Python3 swapcase()方法 | 菜鸟教程

Chat

Docs

Manage the custom docs that you've added.

- Python Indexed 9/11/24, 12:24 PM
- zhipu Indexed 9/11/24, 11:20 AM
- openai Indexed 9/11/24, 11:23 AM
- xunfei-spark Indexed 9/11/24, 11:32 AM
- python3 Indexed 9/11/24, 12:23 PM

Indexed 10 pages

- ZHIPU AI OPEN PLATFORM

Chat

Docs

Manage the custom docs that you've added.

- Python Indexed 9/11/24, 12:24 PM
- zhipu Indexed 9/11/24, 11:20 AM
- openai Indexed 9/11/24, 11:23 AM
- xunfei-spark Indexed 9/11/24, 11:32 AM
- python3 Indexed 9/11/24, 12:23 PM

Indexed 1 pages

- Quickstart tutorial - OpenAI API

Chat

Always search the web

爬取成功之后，就可以作为快捷指令使用了。使用案例，我用 **zhipu** 来演示下。

无论是右侧 Chat 窗口、Composer、还是指定文件内的窗口，都可以使用 @ 调出指令。

The screenshot shows the Feishu interface with a document card for 'zhipu' at the top. A context menu is open over the card, containing options: 'Show Commits & PRs' (with a gear icon), '+ Add new doc', and '@zhipu'. Below the card, the Feishu composer bar is visible with input fields for 'claude-3.5-sonnet', '@ Mention', and 'Image', along with context switches for 'no context', 'chat', and 'codebase'.

注意，建议后缀带上这段话：

"请根据文件内容进行查找，如果找到，引用原文进行回答，并返回引用原文的地址，如果找不到，请回复不知道，一定不要胡编乱造。"

主要目的是：

1. 为了避免大模型捏造一些内容出来。
2. 可以溯源，即使找到内容，也可以通过链接跳转过去确认。

我们可以看下使用的效果：

The screenshot shows a search interface with the following details:

- Search bar: @zhipuai 构建知识库的代码案例是什么, 请根据文件内容进行查找, 如果找到, 引用原文进行回答, 并返回引用原文的地址, 如果找不到, 请回复不知道, 一定不要胡编乱造。
- Below the search bar: `claude-3.5-sonnet`, `@Mention`, `Image`.
- Buttons at the top right: `no context`, `chat`, `codebase`.
- Section labeled `DOCS PAGES` with three entries: `ZHIPU AI OPEN PLATFORM`, `ZHIPU AI OPEN PLATFORM`, `ZHIPU AI OPEN PLATFORM`.
- Text below the results: 对不起,我在提供的文档内容中没有找到关于构建知识库的具体代码案例。这些文档主要涉及了智谱AI的各种模型介绍、使用方法和性能评估等内容,但没有提到知识库构建的相关代码。因此,我无法回答您的问题。如果您需要这方面的信息,可能需要查阅其他相关文档或资料。
- A small icon in the bottom right corner.

剪头指向的就是对应网页内容。

@zhipuai 图片理解示例的代码案例是什么, 请根据文件内容进行查找, 如果找到, 引用原文进行回答, 并返回引用原文的地址, 如果找不到, 请回复不知道, 一定不要胡编乱造。

DOCS PAGES

ZHIPU AI OPEN PLATFORM ZHIPU AI OPEN PLATFORM ZHIPU AI OPEN PLATFORM + 5 pages

根据提供的文档内容, 我找到了zhipuai图片理解示例的代码案例。以下是原文引用:

```
from zhipuai import ZhipuAI

client = ZhipuAI(api_key="") # Fill in your own APIKey
response = client.chat.completions.create(
    model="glm-4v-plus", # Fill in the model name to be called
    messages=[
        {
            "role": "user",
            "content": [
                {
                    "type": "image_url",
                    "image_url": {
                        "url" : "https://www.zhipuai.cn/assets/images/aboutus/company.jpeg"
                    }
                },
                {
                    "type": "text",
                    "text": "What's in the picture?"
                }
            ]
        }
    ]
)
print(response.choices[0].message)
```



这个代码案例展示了如何使用zhipuai库调用GLM-4V-Plus模型来理解图片内容。1

简单总结下, 爬取功能尚不完善, 但可以尝试一下, 说不定你需要的指南能够爬取成功。

3.4 谷歌插件上手 Cursor

适用人群: 没有编程基础的小伙伴

备注: 案例比较简单, 大家仔细阅读 AI 回答的结果即可。

去年 9 月, 我用 ChatGPT 编写过自己的第一个 Chrome 插件: [任何人都可以利用ChatGPT开发自己的谷歌插件](#)

我们来将 ChatGPT 和今天的 Cursor 做个对比。

当时做了两个案例, 第一个: **分割图片为四等份**。第二个: **自动分组浏览器Tab**。

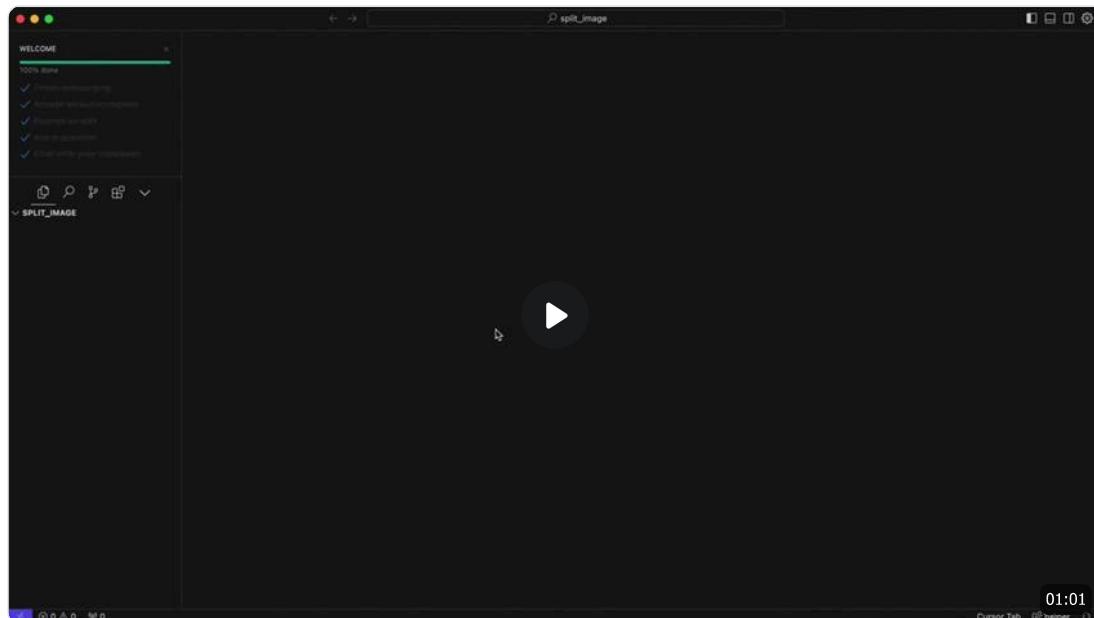
1、案例一: 分割图片为四等份

主要为了快速分割 Midjourney 生成的图片四等份。



当时和 ChatGPT 对话，加上创建文件和部署插件，前前后后花了 15 分钟左右吧，我们来看看 Cursor 的能力。

使用 Cursor 编写代码，加上部署和演示，总共也就一分钟多一点，速度非常快。



Cursor 提示词：

PlainText
请帮我开发一个谷歌插件，用于将用户上传的图片，平均切分为4张图片，并自动下载到本地电脑。
注意使用 manifest v3 版本开发，注意按钮中文编码的问题。

2、案例二：自动分组浏览器 Tab

主要为了解决以下痛点：

1. 浏览器 tab 过多时,容易混乱
2. 无法自动分组, 手动分组麻烦

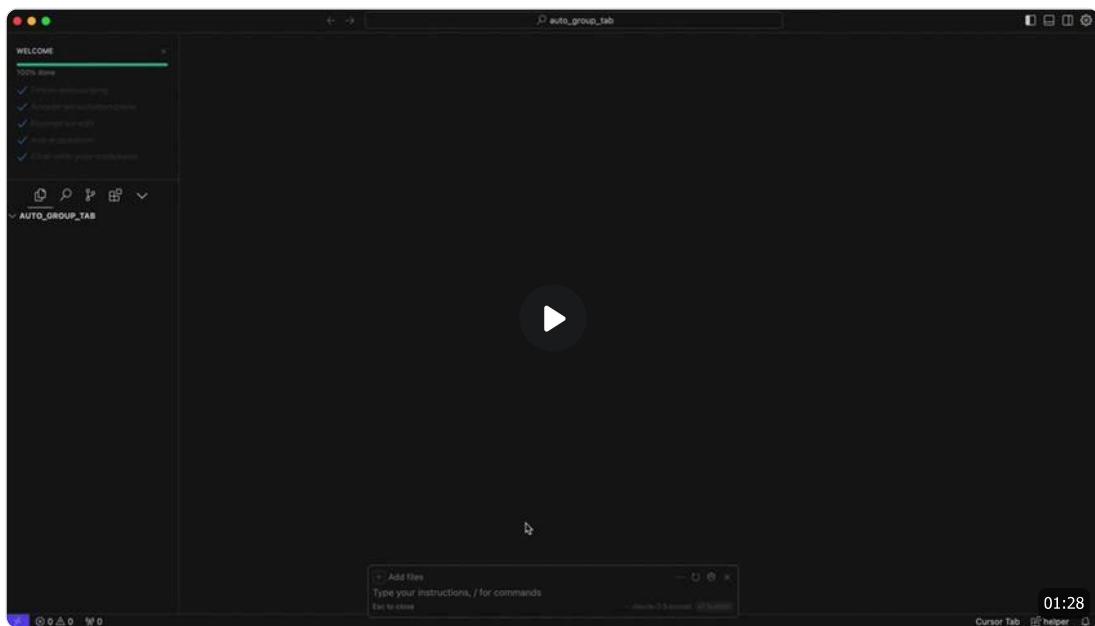


这个案例让我感触颇深。

去年 9 月做这个案例时，我与 AI 对话、调试、纠错、部署，花了 2 个多小时。而这次，用 Cursor 不到五分钟就搞定了，没有做任何纠错，效率提高了二十多倍。

我突然想到：一年前，2 个多小时，我是怎么忍得下来的。

AI 的进步，用一句话来形容：“**AI 一天，人间一年**”。



Cursor 提示词：

Markdown

```
1 # 角色：你是一位资深程序员，能够熟练开发google浏览器插件。
2
3 # 目的：我要开发一个插件，用于将google浏览器的tab自动分组，根据配置的域名自动分组。通过
4 插件，如果用户打开过多的浏览器tab，这样看着会比较混乱。自动分组后，会显得更加整洁和干净。
5 # 功能：
6 1. 点击插件，可以输出需要设置的域名，分组名称，和分组的颜色，颜色有(Blue、Red、Grey、
7 Yellow、Green)
8 2. 点击保存后，页面消失，显示“保存成功”
```

```

8 3. 当在浏览器打开对应域名的链接，后台将自动根据分组名称，和颜色，进行分组。
9 - 若打开新链接时，没有分组，则创建分组
10 - 若打开新链接时，已经存在相同域名的分组，则自动合并
11
12 # 注意：
13 1. 注意使用 manifest v3 版本开发。
14 2. 注意中文编码的问题。

```



【学习安排】跳转入口： 学习安排

AI破局

四、学习使用AI解答编程常见疑问 @雨飞

零基础的人使用AI辅助学习编程，一般有两种不同的方式，一种是使用原生的AI大模型，比如ChatGPT、Claude3.5，智谱清言、deepseek、kimi等；另外一种就是使用常见的一些AI编程工具，它们都用对话功能，比如Cursor、GithubCopilot等，这些编程工具都对代码生成、Bug修复等方面做过深度的优化，通常情况会比直接使用大模型效果要好。

有能力的推荐使用Claude3.5，截止到24年10月份，仍然是最好用的代码大模型。考虑到有些朋友使用不便，也可以使用的国产AI大模型有：

- 1、DeepSeek <https://www.deepseek.com/>
- 2、智谱轻言 <https://chatglm.cn/>
- 3、通义千问 <https://tongyi.aliyun.com/>

不管是原生大模型还是AI编程工具，核心还是需要我们和大模型进行互动，阐述清楚我们的问题，和大模型进行对话的内容，我们一般称为提示词（Prompt），因此掌握基础的提示词编写能力就可以达到辅助编程的目的。而AI编程工具，无非是封装了更多优质的提示词，下面我们将讲一些如何编写高质量的提示词辅助我们编程。

4.1 如何编写提示词

一条优秀的提示词应该清晰明确且具有针对性，能够准确引导模型理解并回应你的问题。

对比下面两条提示词：

- 1) 请帮我实现一个数组排序的算法
- 2) 请帮我用冒泡排序法，利用Python语言实现对数组[19,2,3,5]从大到小的排序

很明显提示词2能更准确的表述我们的问题，并且可以得到更好的回答。

实现编程功能的提示词不需要特别复杂，基本上只要满足「角色」、「背景」、「任务」、「要求」这几个部分就可以生成质量很高的代码。

角色：一般我们都会给大模型限定一个角色，以便它可以更好的理解我们的任务。在AI编程中，常用的角色一般有程序员，软件开发人员，产品经理，测试工程师等。

背景：必要时，可以给大模型提供所需的背景知识，比如需要使用的编程框架，语言。大模型能从你这里获取的，只有你输入的所有提示词，不要假想大模型知道任何事情。

任务：我们要让大模型完成什么样的任务，任务应该是明确，具体，且清晰的，比如利用冒泡排序法对数据进行排序就比对数据进行排序更具体，清晰。

要求：必要时，可以提供其他的要求，比如输出完整的代码，输出JSON格式的数据等。

最后，值得大家注意的是，随着大模型能力的进一步提升，尤其是在数学和编码层面的能力已经达到了一个比较高的水平，我们不需要写很多复杂的提示词就可以得到想要的结果。这种特别复杂的提示词反而可能导致你不知道哪个地方出现问题。随着学习的不断深入，大家对编程的知识有了进一步的了解和认知，可以逐步的学习和优化自己的提示词，锻炼提示词的书写水平。

记住一点，我们使用AI工具辅助编程，**最重要的是帮助我们解决问题，开发出相应的程序**。只要这个提示词能解决你现在的问题，那么这个**提示词就是有价值的**。

4.2 编程常用提示词

Cursor官方为我们提供了一些自己写好的提示词，可以到下面网站去复制学习：

 <https://cursor.directory/>

然后我们介绍一些可以在实际编程中使用的提示词。

1、AI辅助代码生成

- 1 你是一名资深的Python编程工程师，请为以下需求编写代码：
- 2 编程语言： Python
- 3 代码目的： 实现一个简单的GUI计算器
- 4 代码功能： 实现两数的加、减、乘、除运算
- 5 编码要求： 采用面向对象编程方式， GUI使用Tkinter

2、AI辅助代码优化

- 1 你是一位软件开发人员，请根据以下要求整理和优化代码：
- 2 目标： 提高代码的可读性，发现潜在的错误和问题，降低维护成本，提高开发效率
- 3 代码内容： {实际代码}

3、AI辅助解决Bug

- 1 你是一名Bug修复专家，请找出以下代码中的Bug并修复：
- 2 代码： {实际代码}
- 3 目标： 确保代码能够正常运行

- 1 你是一位软件开发人员，下面是我的代码以及出现的问题，请帮我修复代码：
- 2 代码： {实际代码}

3 问题: {错误内容}

4、AI辅助代码注释

- 1 你是一名文档编写专家, 请针对以下代码编写相应的文档和注释。
- 2 代码内容: {实际代码}

5、AI辅助学习

- 1 【角色】充当一名有丰富实战经验的Python工程师
- 2 【背景】我是一个Python的初学者, 想系统的入门和学习Python语言
- 3 【任务】你要和我进行对话, 根据我输入的内容, 去进行讲解和说明
- 4 【要求】我希望你首先可以列举出Python初学者需要掌握的要点, 并注意根据不同经验水平的用户给出不

6、AI辅助生成测试用例

- 1 你是一名测试工程师, 以下是一个功能模块的代码, 请为其设计相关的测试用例。
- 2 代码内容: {代码内容}



【学习安排】跳转入口: ★ 学习安排

AI破局

五、AI编程简单案例

5.0 Python 环境搭建

推荐使用anaconda或者miniconda安装python, 有经验的小伙伴可以跳过。

两者的区别就是anaconda会安装很多的python库, 占据的空间较大, 而miniconda占据的空间较小, 其他方面没有区别。建议是安装miniconda就可以了。

anaconda地址, 可以尝试自己下载安装。

<https://www.anaconda.com/>

一) Miniconda的地址

通常, 使用官方地址下载和安装就可以。如果官方镜像打开有问题, 可以尝试使用清华源的镜像地址。

官方地址:

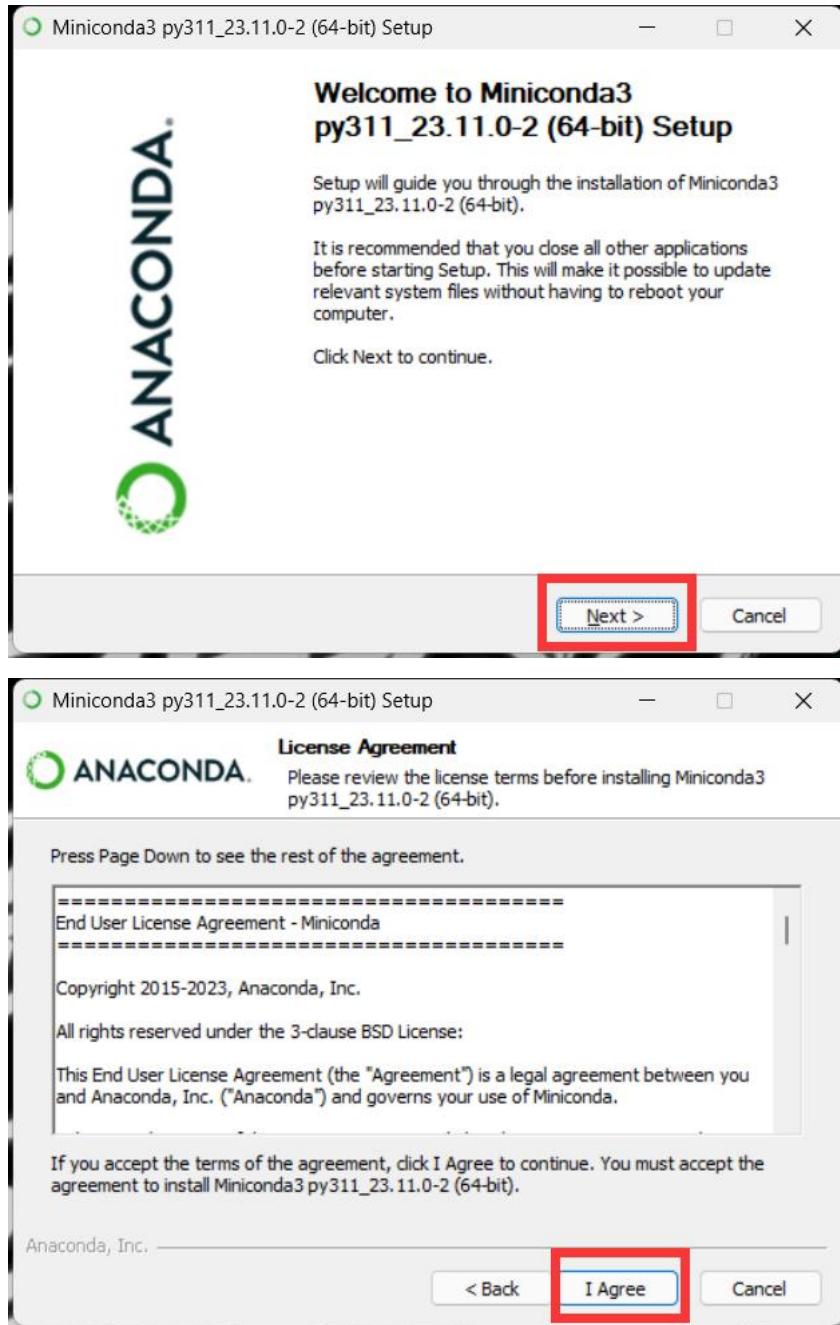
<https://docs.conda.io/en/latest/miniconda.html>

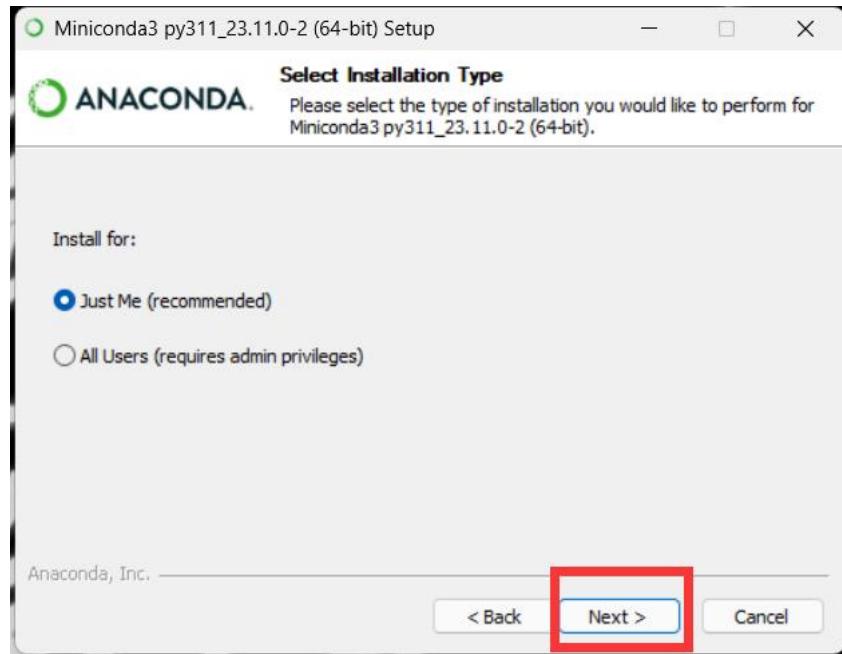
清华源的镜像：

<https://mirrors.tuna.tsinghua.edu.cn/anaconda/miniconda/>

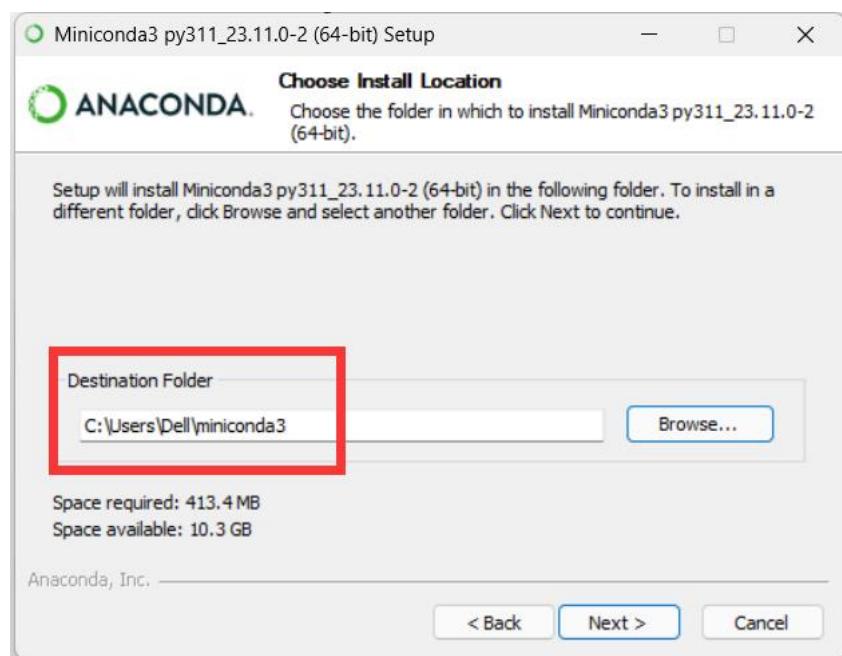
二) Miniconda安装过程

windows下直接点击exe的文件进行安装，和安装软件没有太多区别，下面对需要注意的地方进行说明。

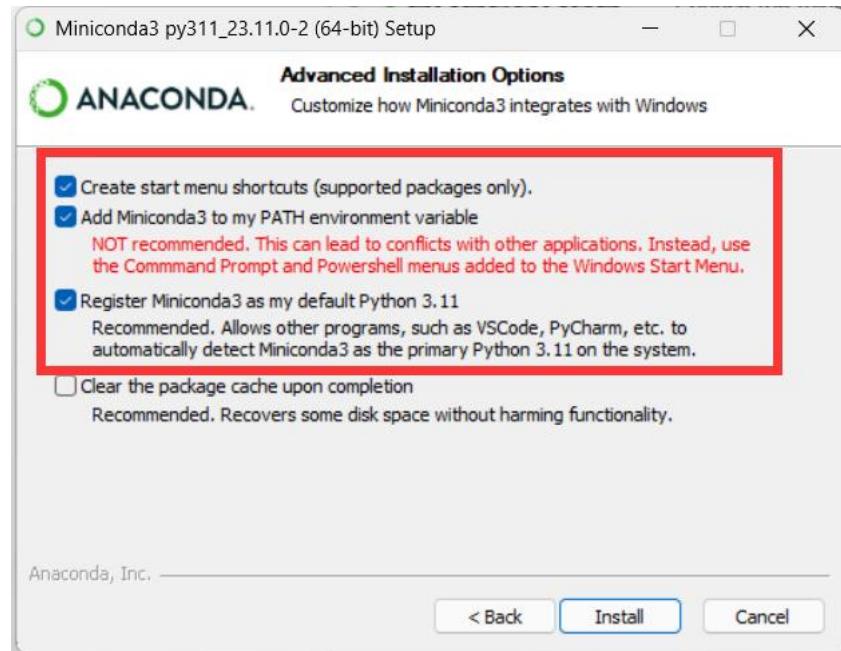




注意，在这一步时，点击右边的**browse**按钮，选择你电脑的其他盘，并且要求**目录不能有中文、空格**。



全部勾选之后，进行安装。



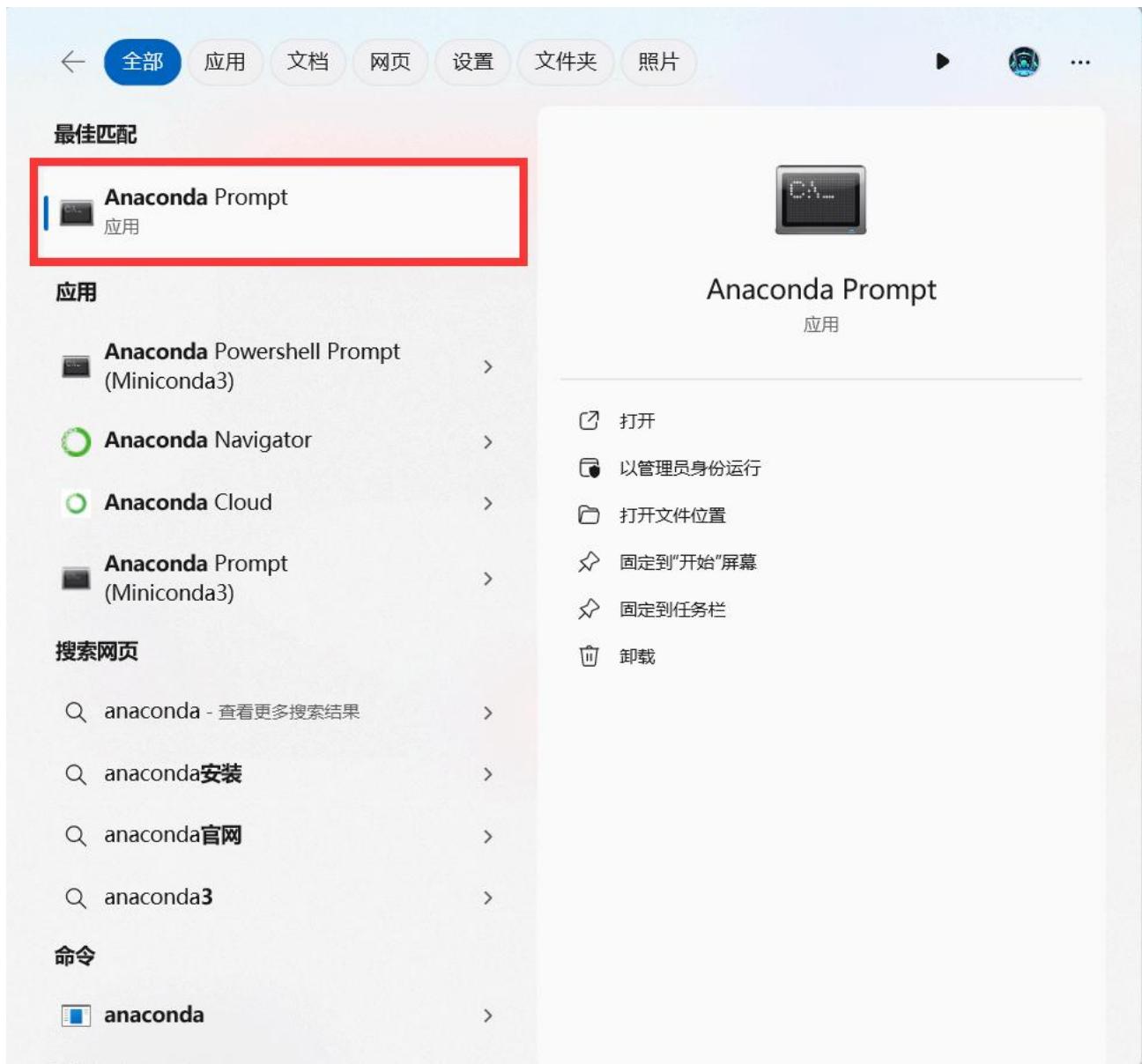
等待安装结束即可。

三) 命令行验证

在桌面上找到搜索框，然后输入anaconda。



在搜索框弹出的地方，如下所示，找到并打开**Anaconda Prompt**。



输入 conda env list，无报错，显示类似下图的内容。

```
Microsoft Windows [版本 10.0.22621.3007]
(c) Microsoft Corporation。保留所有权利。

C:\Users\DELL>conda env list
# conda environments:
#
base * D:\Anaconda3
```

可以使用vscode、pycharm等工具去编写python代码，当然Cursor本身也可以编写python代码。

vscode下载地址：

<https://code.visualstudio.com/>

pycharm下载地址：

<https://www.jetbrains.com/pycharm/>

四) 编写第一行代码

安装好文本编辑器后，输入以下代码：

```
1 print('hello, world')
```

注意 `print` 前面不要有任何空格。然后，选择一个目录，例如 `D:\work`，把文件保存为 `hello.py`，就可以打开命令行窗口，把当前目录切换到 `hello.py` 所在目录。我们使用上面的**Anaconda Prompt**，打开之后，切换到指定目录。

The screenshot shows the Anaconda Prompt window. The command history is as follows:

```
(base) C:\Users\Dell>D:
(base) D:>cd work
(base) D:\work>python hello.py
hello world
(base) D:\work>
```

按截图的步骤输入内容并按回车键执行，即可看到最终的输出。这样就表示，我们的第一个python程序执行成功了。

注意事项

这个文件也可以保存为别的名字，比如 `first.py`，但是必须要以 `.py` 结尾，其他的都不行。此外，文件名只能是英文字母、数字和下划线的组合。

五) 创建和激活一个新环境

默认conda会提供一个base环境，如果我们想新建一个新的环境，只需要打开 **Anaconda Prompt**，使用下面命令创建一个新环境。

```
1 conda create -n rag python=3.10
```

激活一个新环境。

```
1 conda activate rag
```

激活环境之后，Anaconda Prompt 会显示切换到新环境。

六) 安装Python包

使用pip就可以安装指定的包，需要将包名替换为你要安装的包。这个很重要，我们后面就需要安装指定的包。

```
1 pip install 包名
```

也可以使用镜像网址去安装，可以加速安装过程。

```
1 pip install -i https://pypi.tuna.tsinghua.edu.cn/simple/ 包名
```

常见镜像网站：

- 阿里云镜像：<https://mirrors.aliyun.com/pypi/simple/>
- 清华大学镜像：<https://pypi.tuna.tsinghua.edu.cn/simple/>
- 豆瓣镜像：<https://pypi.doubanio.com/simple/>
- 中科大镜像：<https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple/>

5.1 开发小游戏 @雨飞 @加加

我们选择使用 Python 制作小游戏，需要根据前面章节安装好 Python 的开发环境。

1、加减乘除计算器

计算器是一个非常简单，且比较通用的项目，我们思考下计算器有什么功能，并以此编写提示词。

- ```

1 帮我实现一个Python的计算器程序，要求的功能如下:
2 1、能够实现加减乘除四则运算
3 2、提供括号功能，允许用户输入括号表示优先级
4 3、提供删除和清空按钮
5 # 注意
6 - 请注意中文的格式编码问题

```

Cursor 会提示我们需要安装 Python 和 tkinter 库，按照提示的内容装好之后就可以直接运行代码了。

**CHAT    COMPOSER**

Composer • calculator.py ✓

Checkpoint 0 saved. checkout

帮我实现一个Python的计算器程序，要求的功能如下:

- 1、能够实现加减乘除四则运算
- 2、提供括号功能，允许用户输入括号表示优先级
- 3、提供删除和清空按钮

# 注意

- 请注意中文的格式编码问题

好的,我会用中文回答您的问题,并帮您实现一个带有所需功能的Python计算器程序。以下是实现该计算器的代码:

calculator.py 1/1 ✓ → Tab to open

这个程序实现了您要求的所有功能:

1. 能够实现加减乘除四则运算: 程序使用Python的 eval() 函数来计算表达式,支持所有基本的算术运算。
2. 提供括号功能: 用户可以输入左括号 ( 和右括号 ) 来表示运算优先级。
3. 提供删除和清空按钮:
  - C 按钮用于清空显示框
  - Del 按钮用于删除最后一个输入的字符

关于中文格式编码问题,这个程序使用了Python 3,默认支持UTF-8编码,所以不需要特别处理中文字符。所有的中文字串(如"计算器"、"错误"等)都可以直接使用。

要运行这个程序,您需要安装Python和tkinter库(通常Python安装时会默认包含tkinter)。将代码保存为 `calculator.py` 文件,然后在命令行中运行 `python calculator.py` 即可启动计算器程序。

这个计算器提供了一个简单的图形界面,用户可以通过点击按钮或使用键盘输入算式。当点击 = 按钮时,程序会计算结果并显示。如果输入的表达式无效,会显示"错误"。

如果您需要任何进一步的修改或有任何问题,请随时告诉我。

Completed

Reapply all Save all

+ Add files

Edit, refactor, or add code (/ for commands)

Esc to blur

claude-3.5-sonnet Submit

展示效果



## 2、贪吃蛇小游戏

下面我们介绍用提示词实现一个简易的贪吃蛇游戏。

提示词：

- 1 你是一名Python专家, 请编写一段代码:
- 2 项目名称: 贪吃蛇游戏
- 3 # 要求
- 4 1、输出完整的贪吃蛇游戏代码
- 5 2、果实要为黄色, 蛇身为黑色
- 6 3、增加一个计分板功能, 每吃到一个果实得一分

Cursor 会根据我们的要求, 去实现这个功能, 直接接受代码就行。

The screenshot shows a dark-themed AI code editor interface. At the top, there are tabs for 'CHAT' and 'COMPOSER', with 'COMPOSER' being the active tab. To the right of the tabs are buttons for 'Clusters', a plus sign, a refresh icon, a square icon, and a close button. Below the tabs, there are two tabs: 'Composer' and 'snake\_game.py'. A message indicates 'Checkpoint 0 saved. checkout'. A large text box contains the following text:

你是一名Python专家, 请编写一段代码:  
项目名称: 贪吃蛇游戏  
# 要求  
1、输出完整的贪吃蛇游戏代码  
2、果实要为黄色, 蛇身为黑色  
3、增加一个计分板功能, 每吃到一个果实得一分

好的,我会用中文回答您的问题,并为您编写一个符合要求的贪吃蛇游戏代码。以下是完整的Python代码,包含了您要求的所有功能:

`snake_game.py` 1/1 → Tab to open

这段代码实现了一个基本的贪吃蛇游戏,包含以下特性:

1. 完整的贪吃蛇游戏功能
2. 果实为黄色 (YELLOW),蛇身为黑色 (BLACK)
3. 添加了计分板功能,每吃到一个果实得一分

游戏的主要组成部分:

- Snake 类: 用于管理蛇的移动、生长和绘制
- Food 类: 用于生成和绘制食物
- game\_loop() 函数: 包含游戏的主循环,处理事件、更新游戏状态和绘制画面

游戏控制:

- 使用方向键控制蛇的移动
- 吃到食物后,蛇会变长,得分增加
- 撞到墙壁或自己的身体会导致游戏结束

At the bottom of the interface, there are buttons for 'Completed', 'Reapply all', 'Save all', 'Reject all', and 'Accept all' (which is highlighted). There is also a '+ Add context' button, an input field for 'Edit, refactor, or add code (/ for commands)', and status indicators for 'claude-3.5-sonnet' and 'Submit'.

如果遇到了 `ModuleNotFoundError: No module named 'pygame'` 这个问题, 需要按照下面指示的命令, 去安装 pygame 库。

要运行这个游戏,您需要安装 Pygame 库。可以使用以下命令安装:

```
pip install pygame
```

然后,将代码保存为 `snake_game.py` 文件,并运行它即可开始游戏。祝您玩得开心!

如果遇到了其他问题,直接和 Cursor 对话进行修改即可。

只有吃到苹果的时候,蛇身才会增加一个长度。请继续修改代码

理解了,您希望只有在蛇吃到苹果时,蛇身才会增加长度。我们需要修改 `Snake` 类的 `move` 方法和 `game_loop` 函数。以下是修改后的代码:

 snake\_game.py 3/3 ✓

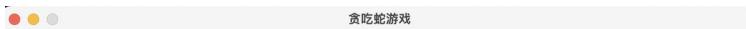
主要的修改包括:

1. 在 `Snake` 类中添加了一个 `grow_flag` 属性,用于标记是否应该增长。
2. 修改了 `move` 方法:
  - 如果 `grow_flag` 为 `False`,则移除蛇尾(保持长度不变)。
  - 如果 `grow_flag` 为 `True`,则不移除蛇尾(增加长度),并将 `grow_flag` 重置为 `False`。
3. 修改了 `grow` 方法,现在它只是将 `grow_flag` 设置为 `True`。
4. `game_loop` 函数中的逻辑保持不变,因为我们只在蛇吃到食物时调用 `grow` 方法。

这些修改确保了蛇只有在吃到苹果(食物)时才会增加长度。在其他情况下,蛇会保持当前长度移动。

这样修改后,游戏的基本逻辑保持不变,但蛇的长度增长现在只发生在吃到食物的时候。您觉得这样的修改是否符合您的要求?

展示效果

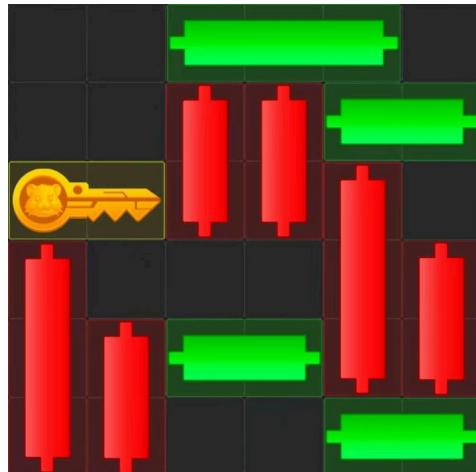


**Score: 3**

### 3、滑块解密游戏（加加）

🎁 起因是我在参与WEB3空投项目hamster的时候，里面有个解密小游戏吸引了我，我想要通过代码把游戏复刻出来，然后再让计算机帮我解密。于是我用cursor帮我解决了这个问题。这里，我抛砖引玉，把生成游戏并自动解题的过程跟大家分享一下。（游戏规则见提示词）

代码位置：[https://github.com/ljl3937/cursor\\_demo/blob/main/jigang\\_prompt/main.py](https://github.com/ljl3937/cursor_demo/blob/main/jigang_prompt/main.py)



#### 安装依赖

由于这是一个游戏程序，因此，我们使用pygame来实现。当然，如果不知道用什么实现，可以直接问cursor，这里为了统一讲解，提示词里指定了使用pygame来实现。

Pygame 是一个用于创建游戏和多媒体应用的 Python 库。

安装方法：

```
1 pip install pygame
```

🎁 tips:

把游戏规则和游戏简介写到 README.md文件中，在提示词中使用，能够有效避免与 cursor 的大量沟通。

REAME.md:

```
1 # 滑块解谜游戏
```

```
2
3 ## 游戏介绍
4
5 这是一个经典的滑块解谜游戏，灵感来源于著名的"华容道"游戏。在这个游戏中，玩家需要通过移动不同的方块，将金色的"钥匙"方块移动到右侧边缘，使其右边缘到达或超过游戏区域的右边界。
6
7 ## 游戏规则
8
9 - 游戏目标：将金色的"钥匙"方块移动到右侧边缘，使其右边缘到达或超过游戏区域的右边界。
10 - 游戏界面：游戏在一个6x6的网格上进行。
11 - 方块类型：
12 - 绿色方块：横向移动，占据2或3个单元格
13 - 红色方块：纵向移动，占据2或3个单元格
14 - 金色方块：钥匙，占据2个单元格，需要移动到出口
15 - 移动规则：
16 - 玩家可以用鼠标拖动方块
17 - 方块只能在其允许的方向上移动（绿色横向，红色纵向）
18 - 方块之间不能重叠
19 - 游戏操作：
20 - 左键点击并拖动方块来移动它们
21 - 按'H'键可以获取提示和启动自动解题功能
22 - 按'R'键可以重置游戏
23
24 ## 特殊功能
25
26 - 自动解题：按'H'键后，游戏会自动每秒移动一次方块，直到解开谜题
27 - 重置游戏：按'R'键可以重置游戏到初始状态
28
29 挑战自我，享受解谜的乐趣吧！
```

## 提示词

```
1 根据 README 中的描述，用pygame写一个游戏，游戏中的色块按照以下排列预置。
2 Block(3, 0, GREEN, 3, 1, 'horizontal'),
3 Block(0, 3, GREEN, 2, 1, 'horizontal'),
4 Block(2, 4, GREEN, 2, 1, 'horizontal'),
5 Block(2, 5, GREEN, 3, 1, 'horizontal'),
6 Block(0, 0, RED, 1, 2, 'vertical'),
7 Block(0, 4, RED, 1, 2, 'vertical'),
8 Block(1, 4, RED, 1, 2, 'vertical'),
9 Block(2, 1, RED, 1, 2, 'vertical'),
10 Block(3, 1, RED, 1, 2, 'vertical'),
11 Block(4, 2, RED, 1, 3, 'vertical'),
12 Block(5, 1, RED, 1, 3, 'vertical'),
13 Block(0, 2, GOLD, 2, 1, 'horizontal')
```

根据 README 中的描述, 用pygame写一个游戏, 游戏中的色块按照以下排列预置。

```
Block(3, 0, GREEN, 3, 1, 'horizontal'),
Block(0, 3, GREEN, 2, 1, 'horizontal'),
Block(2, 4, GREEN, 2, 1, 'horizontal'),
Block(2, 5, GREEN, 3, 1, 'horizontal'),
Block(0, 0, RED, 1, 2, 'vertical'),
Block(0, 4, RED, 1, 2, 'vertical'),
Block(1, 4, RED, 1, 2, 'vertical'),
Block(2, 1, RED, 1, 2, 'vertical'),
Block(3, 1, RED, 1, 2, 'vertical'),
Block(4, 2, RED, 1, 3, 'vertical'),
Block(5, 1, RED, 1, 3, 'vertical'),
Block(0, 2, GOLD, 2, 1, 'horizontal')
```

好的, 我会根据您的要求使用Pygame来创建一个滑块拼图游戏。我将在 `slide_puzzle/test.py` 文件中编写代码。以下是游戏的实现:

`test.py` 1/9 → Tab to open

这段代码实现了一个基本的滑块拼图游戏, 具有以下特点:

1. 使用Pygame创建了一个600x600像素的游戏窗口。
2. 实现了一个 `Block` 类来表示每个滑块, 包括其位置、颜色、大小和方向。
3. 按照您提供的规格创建了12个初始方块。
4. 实现了网格绘制功能, 以便更清晰地显示方块位置。
5. 添加了鼠标交互功能, 允许玩家通过点击和拖动来移动方块。
6. 实现了碰撞检测, 确保方块不会重叠或移出游戏区域。
7. 根据方块的方向(水平或垂直)限制了移动。

要运行这个游戏, 您需要安装Pygame库。然后, 您可以直接运行这个Python文件。

游戏的基本玩法是:

- 点击并拖动方块来移动它们。
- 水平方块只能左右移动, 垂立方块只能上下移动。
- 方块不能重叠或移出游戏区域。

这个实现提供了一个基本的游戏框架。您可以根据需要进一步完善游戏, 例如添加胜利条件、计分系统、关卡设计等。

Completed      Save all      Reapply all

Add files

Type your instructions, / for commands

Esc to blur      claude-3.5-sonnet      Submit

当运行提示词后, 我们会发现还有一些小问题, 比如方块的边框有问题, 如图, 方块没有区分, 方块没有分割, 单元格的划线干扰了视觉。



优化提示词如下:

方块内部没有分割线，只在方块之间显示网格线。这个需求好像没有满足

您说得对，我们确实需要调整一下网格线的绘制方式，以确保只在方块之间显示网格线，而不是在方块内部显示分割线。让我们修改 `draw_grid` 方法来实现这个需求：

`game.py 3/6`

这个新的 `draw_grid` 方法会检查每条线是否位于方块之间，只有在方块之间的线才会被绘制。这样就可以确保方块内部没有分割线，只在方块之间显示网格线。

另外，为了使游戏界面更加美观，我们可以在 `draw` 方法中添加一个金色出口标记：

`game.py 4/6 ✓`

这些修改应该能满足您的需求，使方块内部没有分割线，只在方块之间显示网格线，同时也添加了一个明显的金色出口标记。如果您还有其他需要调整的地方，请随时告诉我。

Checkpoint 4 saved. checkout

`game.py`

方块增加边框

好的，我们可以为每个方块添加一个边框，以使它们更加清晰可见。我们将修改 `Block` 类的 `draw` 方法来实现这一点。以下是修改后的代码：

`game.py 5/6 ✓`

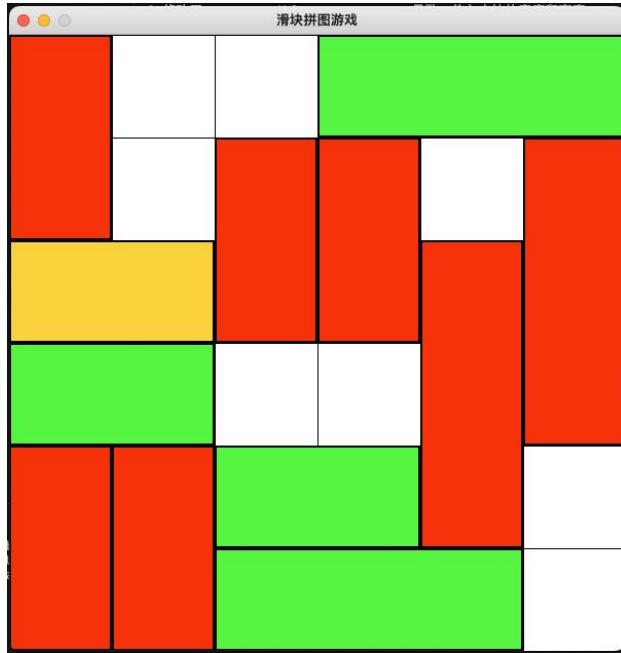
这个修改会为每个方块添加一个黑色的边框。如果你想要不同的边框颜色，可以将 `BLACK` 替换为其他颜色。

另外，为了使金色出口标记更加明显，我们也可以为它添加一个边框。修改 `Game` 类的 `draw` 方法如下：

`game.py 6/6 ✓`

其他的优化过程类似，这里就不再赘述，因为cursor生成的代码会有一定的差异，因此，每个人遇到的问题不同。

最终的效果如下：



## 5.2 开发谷歌插件/爬取资源 @雨飞

谷歌插件通常是指为谷歌浏览器（Google Chrome）开发的扩展程序，可以增强谷歌浏览器功能的小型软件。在AI工具出现之前，我们要制作一款谷歌插件，需要学习JS、HTML等相关的前端知识，还需要了解谷歌插件的开发规范。

而Cursor等AI编程工具的出现，就使得我们通过和AI对话就可以完成插件的制作和开发。下面我们展示几个基础的插件制作和使用过程。

代码位置：

<https://github.com/Xls1994/cursor-code>

所有的代码都可以在这里找到，我们更推荐你亲自尝试编写提示词，而不是简单的复制粘贴代码。

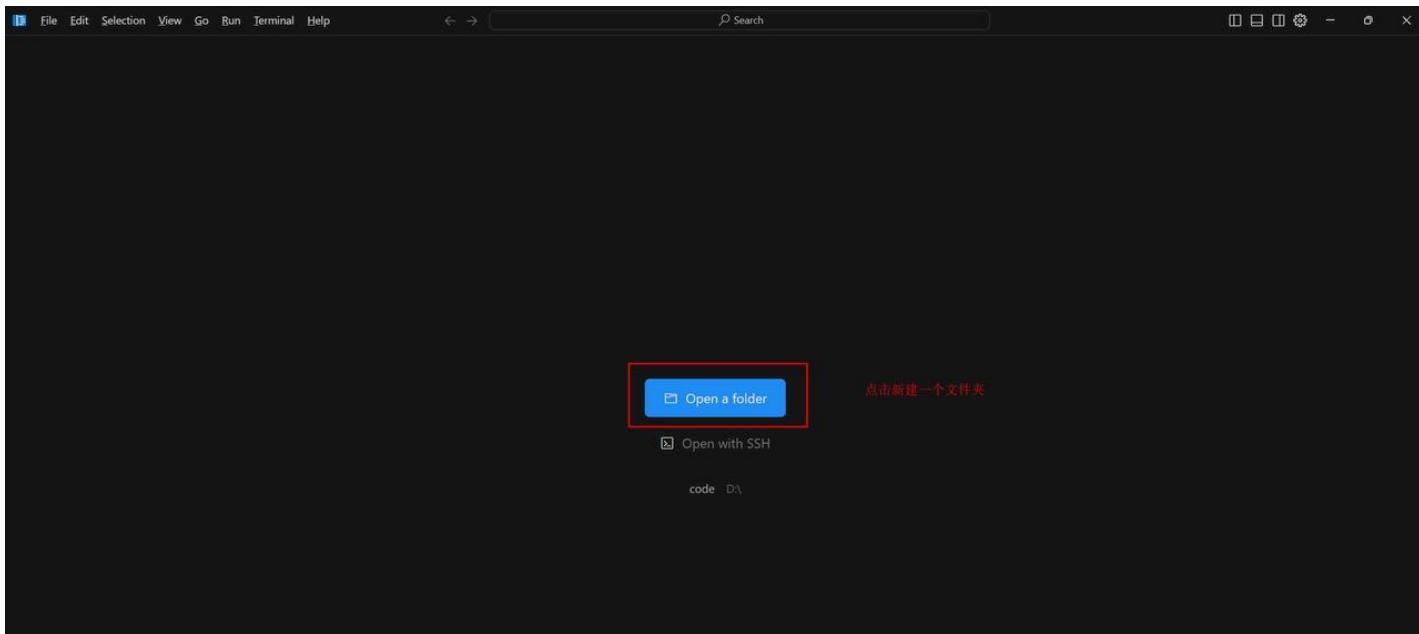
## 1、侧边栏AI助手

简单来说，就是在我们浏览器的右侧展示一个AI大模型的页面，方便我们一边阅读网页，一边调用大模型进行对话。

提示词：

```
1 帮我实现一个 Chrome 插件，要求的功能如下：
2 1、点击插件时，打开右侧边栏，展示网站: https://chat.deepseek.com/
3
4 # 注意
5 - 无需使用图标
6 - 使用V3版本的manifest.json
7 - 请注意中文的格式编码问题
```

1、打开Cursor，新建一个文件夹，这里我们给文件夹起名为test。



2、然后使用 Ctrl+I 打开 Composer，然后在对话框中输入上面的提示词，按回车即可。Composer会自己去创建一些代码文件，**最后会出现一个「Accept all」选项，需要点击接受所有的代码**。注意，**每次 Cursor 输出的内容并不会完全一致，只要最终能实现我们的需求就可以**。

**CHAT**    **COMPOSER**

Composer • manifest.json • background.js • sidepanel.html •

Checkpoint 0 saved. checkout

帮我实现一个 Chrome 插件，要求的功能如下：

- 1、点击插件时，打开右侧边栏，展示网站：<https://chat.deepseek.com/>

# 注意

- 无需使用图标
- 使用V3版本的manifest.json
- 请注意中文的格式编码问题

好的，我来帮您实现这个Chrome插件。我会使用Manifest V3版本，并注意中文的格式编码问题。以下是实现这个插件所需的文件和代码：

首先，我们需要创建一个 `manifest.json` 文件：

{ manifest.json 1/2 • → Tab to open

接下来，我们需要创建 `sidepanel.html` 文件：

△ sidepanel.html 1/1 •

最后，我们需要创建一个 `background.js` 文件来处理插件的点击事件：

JS background.js 1/1 •

现在，让我们更新 `manifest.json` 文件以包含 `background.js`：

{ manifest.json 2/2 •

这些文件构成了一个完整的Chrome插件，可以实现您要求的功能：

1. 当用户点击插件图标时，会打开右侧边栏。
2. 侧边栏中会显示 <https://chat.deepseek.com/> 网站。

使用说明：

1. 创建一个新文件夹将上述所有文件放入其中。
2. 在Chrome浏览器中，进入 `chrome://extensions/` 页面。
3. 开启右上角的“开发者模式”。
4. 点击“加载已解压的扩展程序”，选择包含这些文件的文件夹。

插件安装完成后，您可以点击工具栏中的插件图标来打开DeepSeek Chat侧边栏。

请注意，所有文件都使用UTF-8编码保存，以确保中文字符能够正确显示。如果您在使用过程中遇到任何问题或需要进一步的修改，请随时告诉我。

Completed

+ Add context

Edit, refactor, or add code (/ for commands)

Esc to blur

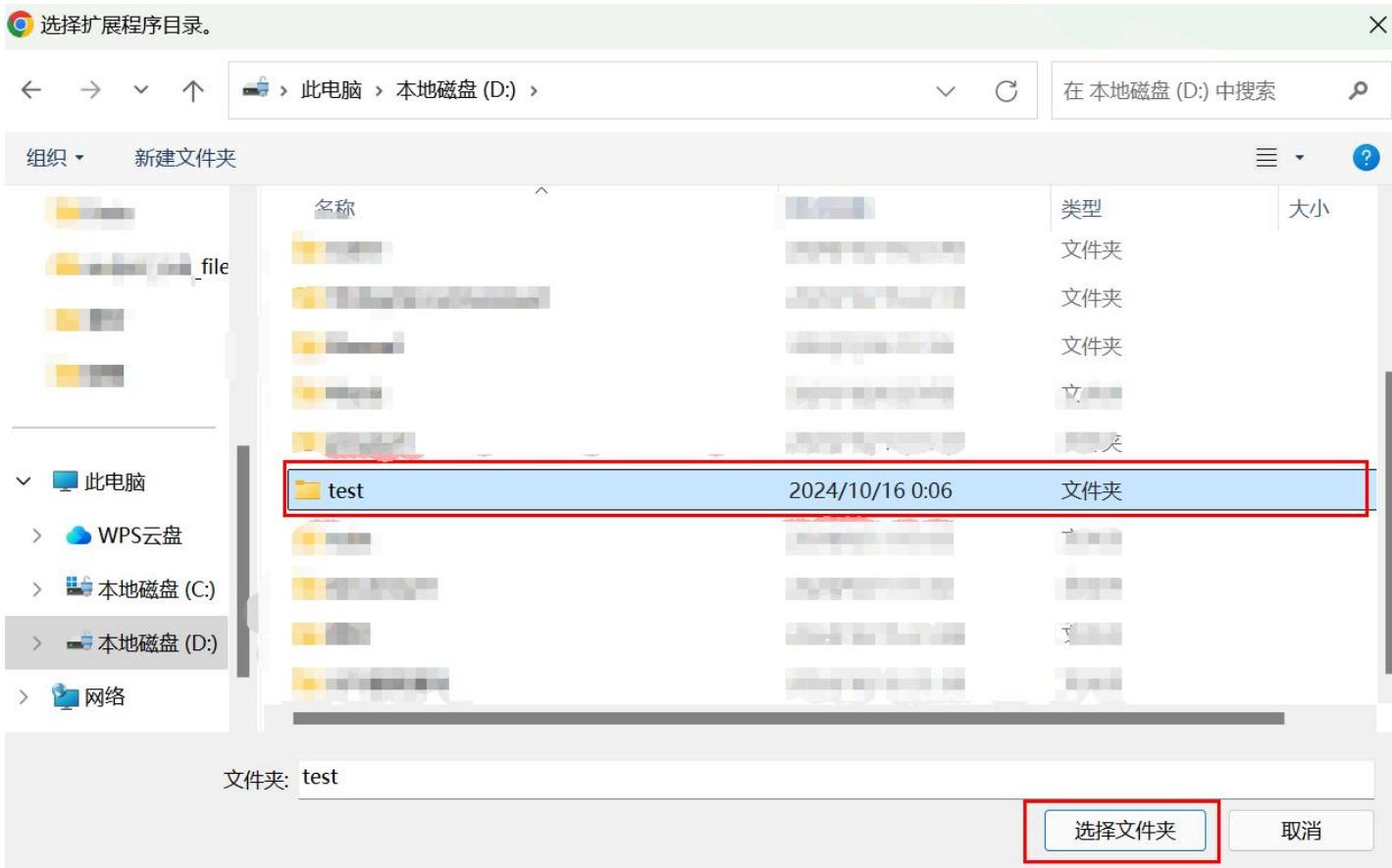
Reapply all Save all Reject all **Accept all**

▼ claude-3.5-sonnet Submit

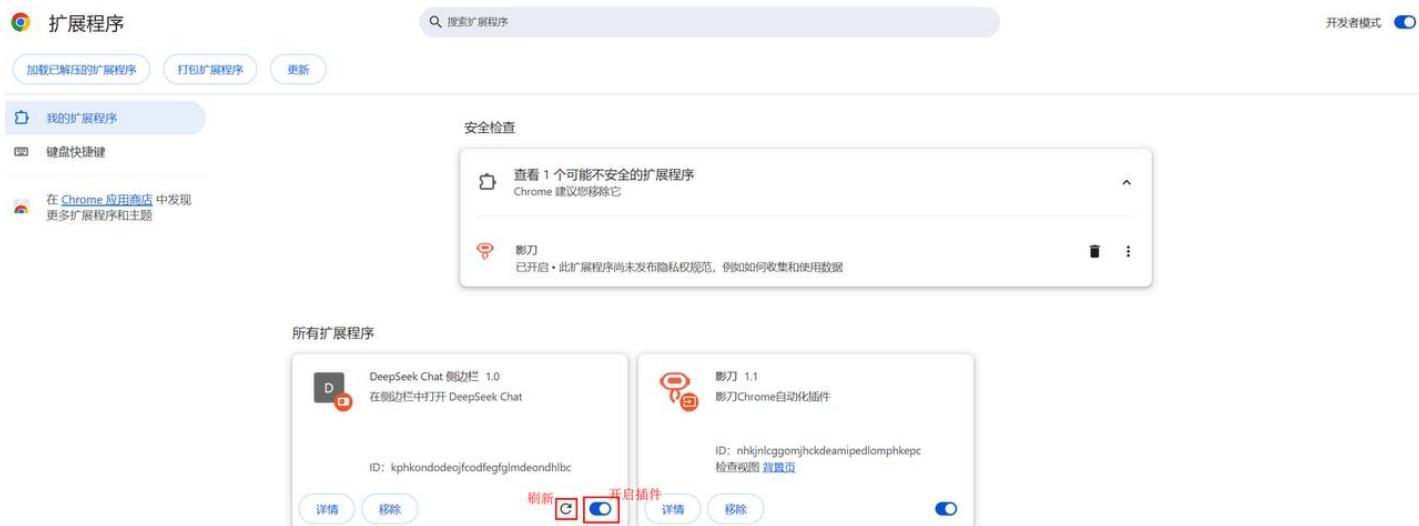
3、在谷歌浏览器输入：chrome://extensions/，打开「开发者模式」，如下图所示，右上角按钮显示出蓝色。



4、点击「加载已解压的扩展程序」，在弹出的对话框中选择我们之前的文件夹。



当我们更新代码后，需要点击左侧刷新按钮之后再测试。



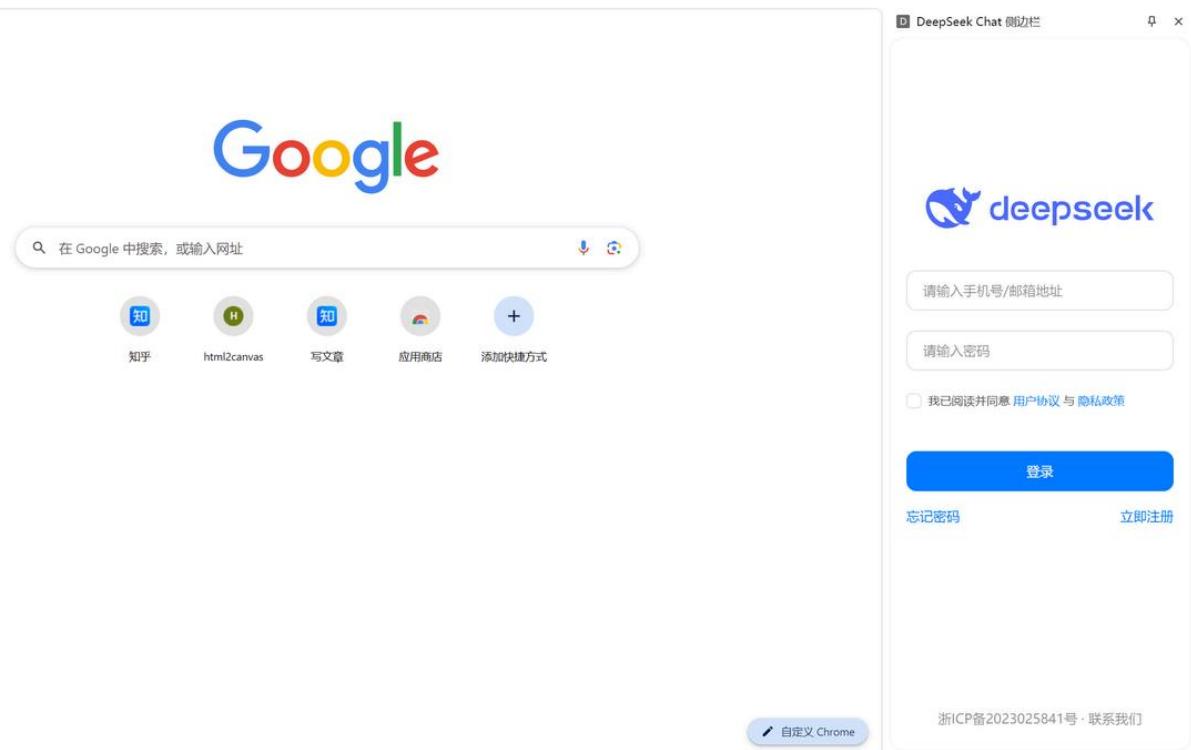
6、在主界面找到对应的插件，点击进行测试。



后续所有的谷歌插件都可以按照 3-6 的步骤进行开发和测试，不再赘述。

### 实际效果

出现下面效果，则表示制作成功。



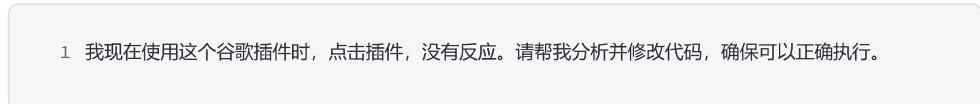
编程过程中出现问题怎么办？

有的时候，很不幸，我们在开发的时候会遇到各种各样的问题，比如下面这种样子，点击插件没有反应了。



我们需要使用 **Ctrl+I** 打开 Composer，勾选好所有的文件，去输入下面的提示词，描述清楚我们的问题。如果，你知道是哪个文件出了问题，就只勾选那个文件。

提示词：



修改之后，点击「Accept all」，然后去刷新 Chrome 插件。



正常修改完代码之后，点击该插件可以正常执行。如果出现问题，需要把出现的错误发送给 Cursor，让它进行后续的修改即可。

和手册截图不一致怎么办？

保持心态，AI 输出的结果本身就有一定的随机性。我们最终的目的是完成自己的应用，AI 输出的回答不一样没有关系。

**但是不管 AI 输出什么内容，我们一定要认真阅读，确保自己都理解之后再去执行后续的操作。**

提示词有没有必要完全参考手册？

**没有必要，我们编写提示词的目的是为了解决问题，你可以根据实际修改和优化提示词。**

当然，也可以参考 Cursor 官方提供的一些高质量提示词：<https://cursor.directory/>

## 2、网站二维码

可以在网页上生成一个二维码，扫码就可以跳转到当前页面。

提示词：

```
1 帮我实现一个 Chrome 插件，要求的功能如下：
2
3 1、打开任意网页时，在右下角显示网页的 logo 图标，点击这个图标时会展开显示网页二维码
4 2、二维码大小256*256、白底，上边距、左右边距都是25px
5 3、二维码的下面分两行显示网站名称和网页标题，标题最多显示10个字
6 4、二维码中间显示网页logo
7
8 #注意
9 - 无需使用图标
10 - 使用V3版本的manifest.json
11 - 请注意中文的格式编码问题
```

和之前操作类似，点击「Accept all」，注意，Cursor 的回答中需要我们去下载一个插件，并放在插件的目录中。

CHAT    COMPOSER

Composer • # styles.css • JS content.js • {} manifest.json •

Checkpoint 0 saved, checkout

帮我实现一个 Chrome 插件，要求的功能如下：

1、打开任意网页时，在左下角显示网页二维码  
 2、二维码大小256\*256、白底，上边距、左右边距都是25px  
 3、二维码的下面分两行显示网站名称和网页标题，标题最多显示10个字  
 4、二维码中间显示网页logo

# 注意：  
 - 无需使用图标  
 - 使用V3版本的manifest.json  
 - 请注意中文的格式编码问题

好的,我来帮您实现这个Chrome插件。我们需要创建几个文件来完成这个功能。以下是实现步骤:

1. 创建manifest.json文件
2. 创建content.js文件
3. 创建qrcode.min.js文件(我们将使用qrcode.js库来生成二维码)
4. 创建styles.css文件

让我们逐个创建这些文件:

1. manifest.json
2. content.js
3. qrcode.min.js

请从 <https://raw.githubusercontent.com/davidshimjs/qrcodejs/master/qrcode.min.js> 下载qrcode.min.js文件并保存在插件目录中

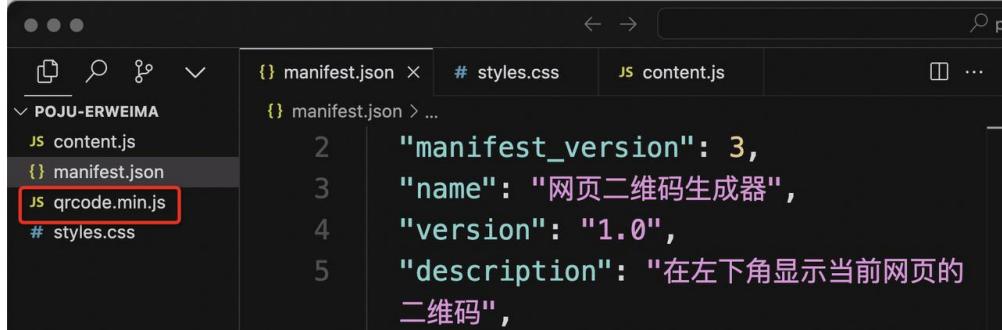
Completed    Reapply all    Save all    Reject all    **Accept all**

+ Add context

Esc to blur    claude-3.5-sonnet    Submit

切记，qrcode.min.js 文件一定要放在对应目录下，可以去下面这个链接下载文件：

<https://raw.githubusercontent.com/davidshimjs/qrcodejs/master/qrcode.min.js>

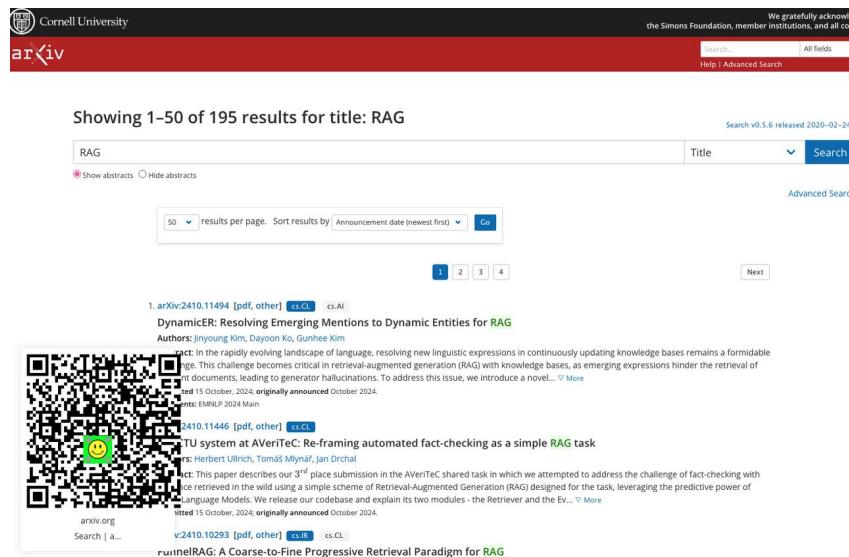


```

{
 "manifest_version": 3,
 "name": "网页二维码生成器",
 "version": "1.0",
 "description": "在左下角显示当前网页的二维码",
}

```

添加谷歌插件，进行测试。正常刷新网页后，会在左侧出现网站二维码。



可以进一步和 Cursor 提要求，对二维码进行美化和调整。

输入提示词：

1 请帮我隐藏掉二维码，只展示网页 logo。当点击 logo 时，才展示二维码。

Cursor 会继续输出并修改代码，如果有问题可以继续对话进行修改和调整。

Checkpoint 1 saved. checkout

请帮我隐藏掉二维码，只展示网页 logo。当点击 logo 时，才展示二维码。

好的，我理解您想要修改插件的功能，使二维码默认隐藏，只显示网页logo，点击logo时才展示二维码。我们需要修改content.js和styles.css文件来实现这个功能：以下是修改后的代码：

1. 修改content.js文件：

```
JS content.js 2/2 •
```

2. 修改styles.css文件：

```
styles.css 2/2 •
```

这些修改实现了以下功能：

1. 默认只显示网页logo，大小为64x64像素。
2. 点击logo时，展开显示二维码、网站名称和页面标题。
3. 再次点击logo时，隐藏二维码、网站名称和页面标题，只显示logo。

请用这些代码替换原来的content.js和styles.css文件。这样，插件就会默认只显示网页logo，点击logo时才展示完整的二维码信息。

Completed Reapply all Save all Reject all Accept all

+ Add context Edit, refactor, or add code (/ for commands)

Esc to blur claude-3.5-sonnet Submit

最终效果。



### 3. 翻译插件

我们继续尝试，实现两种不同的翻译插件。

最简单的一种，就是选取当前的文字，在右键菜单中选择选项，然后跳转到谷歌翻译的页面，然后实现翻译。

根据我们的想法编写提示词：

- 1 帮我实现一个 Chrome 插件，要求的功能如下：
- 2 1、当我们选中网页上的文字并右键时，在菜单中显示“使用谷歌进行翻译”选项，点击该选项后弹出谷歌翻译的界面。
- 3 2、无需使用图标
- 4 3、使用V3版本的manifest.json
- 5 4、请注意中文的格式编码问题

Cursor 会直接根据我们的要求，去创建相关文件，点击「Accept all」，然后去部署好谷歌插件测试下功能。

**CHAT** **COMPOSER**

Composer • manifest.json ✓ background.js ✓

Clusters X

Checkpoint 0 saved. checkout

帮我实现一个 Chrome 插件，要求的功能如下：

- 当我们选中网页上的文字并右键时，在菜单中显示“使用谷歌进行翻译”选项，点击该选项后弹出谷歌翻译的窗口并将选中文字翻译成英文
- 无需使用图标
- 使用V3版本的manifest.json
- 请注意中文的格式编码问题

好的，我来帮您实现这个Chrome插件。我们需要创建几个文件来完成这个功能。让我们一步步来：

- 首先，创建manifest.json文件：

{ manifest.json 1/1 ✓ }

- 接下来，创建background.js文件：

js background.js 1/1 ✓

这些文件就足够实现您要求的功能了。让我解释一下每个文件的作用：

- manifest.json：
  - 使用manifest\_version 3
  - 定义了插件的基本信息
  - 请求了contextMenus权限，这是为了创建右键菜单项
  - 指定background.js作为service worker
- background.js：
  - 在插件安装时创建一个名为“使用谷歌进行翻译”的右键菜单项
  - 当用户点击这个菜单项时，会打开一个新标签页，跳转到谷歌翻译页面
  - 使用encodeURIComponent()函数来处理选中文本，解决中文编码问题

Completed Reapply all Save all

+ Add files Edit, refactor, or add code (/ for commands) claude-3.5-sonnet Submit

Esc to blur

和之前操作类似，点击「Accept all」之后，将插件安装到谷歌浏览器中，并进行测试。

展示结果如下：

选中任意一段文字，右键，在菜单中选择「使用谷歌进行翻译」然后会跳转到谷歌翻译的页面，并将内容翻译成英文。



这一种方式实现非常简单，但是不够丝滑。我们希望当鼠标选中文字的时候就可以进行翻译，不需要额外点击右键菜单。根据我们的需求，编写和优化提示词。

- 1 帮我实现一个 Chrome 插件，要求的功能如下：
- 2 1、当我们选中网页上的文字，然后自动将选中的文字翻译成英文。
- 3 2、在鼠标停留位置展示翻译的结果
- 4 3、结果展示后，点击鼠标左键或者等待 3s 后，自动关闭翻译结果
- 5
- 6 # 注意
- 7 - 无需使用图标
- 8 - 使用V3版本的manifest.json
- 9 - 请注意中文的格式编码问题

### Cursor 输出内容示意图

The screenshot shows the Feishu Composer interface with the following sections:

- CHAT** and **COMPOSER** tabs are visible at the top.
- Clusters** tab is on the right.
- Files** section: manifest.json (1/1), background.js (1/1), content.js (1/1).
- Message** section: "Checkpoint 0 saved. checkout".
- Content Area** (highlighted in yellow):
  - Text: "帮我实现一个 Chrome 插件，要求的功能如下：" followed by a list of requirements.
  - Text: "# 注意" followed by notes.
  - Text: "好的,我理解您想要实现一个Chrome插件,具有以下功能:" followed by a numbered list of requirements.
  - Text: "我会使用Manifest V3来实现这个插件。让我们开始逐步修改和创建必要的文件。"
  - Text: "首先,让我们更新 manifest.json 文件:" followed by a manifest.json file preview.
  - Text: "接下来,让我们修改 background.js 文件:" followed by a background.js file preview.
  - Text: "现在,让我们创建 content.js 文件来实现主要功能:" followed by a content.js file preview.
  - Text: "这些修改实现了您要求的功能:" followed by a numbered list of implementation details.
- Bottom Bar**:
  - Completed
  - Reapply all Save all
  - + background.js
  - Edit, refactor, or add code (/ for commands)
  - Esc to blur
  - claudie-3.5-sonnet Submit

### 展示结果

但是随着预训练语言模型越来越大，这个范式存在以下问题：

- 当我们finetune大模型时，由于训练成本太高，不太可能重新训练所有模型参数
- 以前的方法（论文发表于2021年）都或多或少有其它性能问题，如adapter增加了模型层数，引入了额外的推理延迟；prefix-tuning比较难训练，效果参差不齐。Previous methods (paper published in 2021) all had more or less additional performance issues based on the above background, the paper authors benefited from some findings about intrinsic dimension (intrinsic dimension) of models: models are overparameterized, they have smaller intrinsic dimensions, and model mainly relies on this low intrinsic dimension (low rank) to do task adaptation. Assuming that the change in weight during task adaptation is low rank (low rank) of the model, this leads to the low rank self-adaptation of the model.

林诗栋不敌张本智和获男单亚军

韩军公布朝鲜炸毁朝韩间道路视频

为何老小区加装电梯虎头蛇尾了

张本智和

## 4、爬虫插件

好了，学了这么些插件，我们再做点有意思的。

来试下，使用 Cursor 去定制一个浏览器插件去爬取一些文章，我们这里举一个爬取公众号文章的例子。

提示词：

```

1 帮我实现一个 Chrome 插件，要求的功能如下：
2 1、分析这个网站链接的内容 https://mp.weixin.qq.com/s/yLqUCTKX4tajf4HF7-AbqQ
3 2、基于分析的结果，找到正文部分，并爬取保存到文件中。文件名为 weixin_{拼接现在的时间戳}.tx
4 # 注意
5 - 无需使用图标
6 - 使用V3版本的manifest.json
7 - 请注意中文的格式编码问题

```

Cursor 会直接根据我们的要求，去创建相关文件，点击「Accept all」，然后去部署好谷歌插件测试下功能。

**CHAT**    **COMPOSER**

Clusters + ( ) ×

Composer • JS background.js • JS popup.js • <popup.html• { manifest.json

Checkpoint 0 saved. checkout

帮我实现一个 Chrome 插件, 要求的功能如下:

- 分析这个网站链接的内容  
<https://mp.weixin.qq.com/s/yLqUCTKX4tajf4HF7-AbqQ>
- 基于分析的结果, 找到正文部分, 并爬取保存到文件中。文件名为 weixin\_{拼接现在的时间戳}.txt

# 注意

- 无需使用图标
- 使用V3版本的manifest.json
- 请注意中文的格式编码问题

好的,我来帮您实现这个Chrome插件。我们将创建一个插件,它可以分析微信公众号文章并保存正文内容。以下是实现步骤:

- 首先,创建manifest.json文件:  
{} manifest.json 1/1
- 创建popup.html文件:  
<popup.html 1/1
- 创建popup.js文件:  
JS popup.js 1/1
- 创建background.js文件:  
JS background.js 1/1

这个Chrome插件的工作原理如下:

- manifest.json文件定义了插件的基本信息和权限。
- popup.html和popup.js创建了一个简单的弹出界面,用户可以点击按钮来保存文章内容。

Completed    Reapply all    Save all    Reject all    Accept all

+ Add context    Edit, refactor, or add code (/ for commands)

Esc to blur    claude-3.5-sonnet    Submit

如果遇到了问题, 可以根据报错继续和Cursor对话, 然后让他修改即可。

Uncaught (in promise) Error: Could not establish connection. Receiving end does not exist.

这个错误通常发生在content script尚未加载完成时就尝试向其发送消息。我们需要修改代码以确保content script已经加载并准备好接收消息。让我们对代码进行以下修改:

- 首先, 我们需要在manifest.json中添加content\_scripts配置:  
{} manifest.json 2/4
- 然后, 我们需要修改popup.js文件, 使用chrome.scripting.executeScript来确保content script已经加载:
- 现在我们可以简化content.js文件, 因为主要的提取逻辑已经移到了popup.js中:
- background.js文件保持不变。

这些修改应该能解决"Receiving end does not exist"的错误。现在, 当用户点击"提取文章内容"按钮时, chrome.scripting.executeScript会确保在当前标签页中执行extractContent函数, 无论content script是否已经加载。

请注意, 您需要在manifest.json中添加"scripting"权限才能使用chrome.scripting API:  
{} manifest.json 3/4

这些更改应该能解决连接问题, 并使您的Chrome扩展程序正常工作。如果您还遇到任何其他问题, 请随时告诉我。  
Checkpoint 2 saved. checkout

Error in event handler: TypeError: URL.createObjectURL is not a function at chrome-extension://pmpkbcnfbfmebmfpbofgifddmldnomdi/background.js:5:16

这个错误表明在background.js中使用的URL.createObjectURL函数不可用。这是因为在Manifest V3的Service worker中, 某些Web API (包括URL.createObjectURL) 是不可用的。我们需要修改background.js的代码来解决这个问题。

让我们使用chrome.storage API来临时存储内容, 然后在popup中处理下载。以下是修改后的代码:

- 首先, 修改manifest.json以添加storage权限:  
{} manifest.json 4/4
- 修改background.js文件:  
JS background.js 2/2
- 修改popup.js文件以处理下载:  
JS popup.js 3/3

这些修改解决了在background.js中使用URL.createObjectURL的问题。现在, 内容首先被保存到chrome.storage中, 然后在popup.js中创建Blob和下载。

请注意, content.js和popup.html保持不变。

这些更改应该能解决URL.createObjectURL的问题, 并使您的Chrome扩展程序正常工作。如果您还遇到任何其他问题, 请随时告诉我。

遇到错误时，可以查看相应的谷歌插件，会显示类似下面的内容，点击之后，可以看到错误详情。



从报错的详情中，我们可以清楚的看到报错原因，以及所在的代码位置。

```

1 document.getElementById('saveArticle').addEventListener('click', async () => {
2 const [tab] = await chrome.tab.query({ active: true, currentWindow: true });
3 chrome.scripting.executeScript({
4 target: { tabId: tab.id },
5 function: saveArticleContent
6 });
7 }
8
9 function saveArticleContent() {
10 const content = document.querySelector('#js_content').innerText;

```

将上面涉及到的文件、代码片段输入给 Cursor，并提问，让它帮助我们解决问题。

提示词：

```

1 Uncaught (in promise) TypeError: Cannot read properties of undefined (reading
2 >>>>
3 选中的代码出现了上面这个错误，请修改并给出正确代码。

```

popup.html    JS popup.js    JS popup.js (2-2)

Uncaught (in promise) TypeError: Cannot read properties of undefined (reading 'query')

>>>>

选中的代码出现了上面这个错误，请修改并给出正确代码。

我理解了，这个错误是由于 `chrome.tab` 应该是 `chrome.tabs`。我们来修正这个问题。

以下是修改后的 `popup.js` 文件内容：

JS popup.js 2/2 •

主要的修改是在第二行，将 `chrome.tab.query` 改为 `chrome.tabs.query`。这是因为 Chrome 扩展 API 中使用的是复数形式 `tabs`，而不是单数形式 `tab`。

这个修改应该能解决 "Cannot read properties of undefined (reading 'query')" 错误。错误发生是因为 `chrome.tab` 是未定义的，所以无法读取其 `query` 属性。

使用正确的 `chrome.tabs.query` 方法，我们就可以正确地获取当前活动标签页的信息了。

请用这个更新后的代码替换原来的 `popup.js` 文件内容。这应该能解决你遇到的问题。如果你还有其他问题或需要进一步的帮助，请随时告诉我。

Completed    Reapply all    Save all    Reject all    Accept all

+    < popup.html File ×    JS popup.js File ×

修改完代码之后，先点击右上角的 **全部清除**，然后重新加载 Chrome 插件。如果还有问题，则重复上述的修改步骤。

错误    全部清除

Uncaught (in promise) TypeError: Cannot read properties of undefined (reading 'query')

上下文  
popup.html

堆叠追踪  
popup.js:2 (匿名函数)

```

1 document.getElementById('saveArticle').addEventListener('click', async () => {
2 const [tab] = await chrome.tab.query({ active: true, currentWindow: true });
3 chrome.scripting.executeScript({
4 target: { tabId: tab.id },
5 function: saveArticleContent
6 });
7 }
8
9 function saveArticleContent() {
10 const content = document.querySelector('#js_content').innerText;

```

## 展示效果

正常情况，点击该插件会出现如下图所示的一个按钮，点击按钮会生成一个包含网页正文的文本文档。



## 5、爬取微博热榜

我们也可以使用 python 程序去完成网站爬取的工作，下面简单介绍下使用 Python 的 requests 库去制作一个简单的微博热榜爬取。

提示词：

- 1 请帮我编写一个 Python 程序，去爬取微博的热榜。
- 2 # 要求：
- 3 1、使用 requests 和 python3
- 4 2、目标网站： <https://tophub.today/n/KqndgxeLl9>
- 5 3、爬取表格中热榜的序号、名字、热度、链接，并保存到 csv 文件里。
- 6
- 7 请一步步思考，你可以先打开目标网站分析其网页结构，并编写代码。

需要根据Cursor 的指示，安装对应的 requests 库和 BeautifulSoup 库。

CHAT    COMPOSER    Clusters + ⚙️ ☰ ×

Composer • weibo\_hot\_topics.py •

Checkpoint 0 saved. checkout

请帮我编写一个 Python 程序，去爬取微博的热榜。  
# 要求：  
1、使用 requests 和 python3  
2、目标网站： <https://tophub.today/n/KqndgxeLl9>  
3、爬取表格中热榜的排名、标题、热度和链接，并保存到 csv 文件里。  
请一步步思考，你可以先打开目标网站分析其网页结构，并编写代码。

好的,我会用中文回答您的问题,并按照您的要求编写一个Python程序来爬取微博热榜。让我们一步步来分析和编写代码。

首先,让我们分析目标网站的结构:

1. 打开 <https://tophub.today/n/KqndgxeLl9>
2. 网页中包含一个表格,其中包含了微博热榜的信息
3. 表格的每一行包含排名、标题、热度和链接

基于这个分析,我们可以使用 requests 库来获取网页内容,然后使用 BeautifulSoup 来解析 HTML 并提取所需的信息。最后,我们将使用 csv 模块将数据保存到 CSV 文件中。

让我们开始编写代码:

weibo\_hot\_topics.py 1/2 ↗ Tab to open

这段代码完成了以下任务:

1. 使用 requests 库获取目标网页的内容。
2. 使用 BeautifulSoup 解析 HTML 并提取所需的信息(排名、标题、热度和链接)。
3. 将提取的信息保存到一个列表中。
4. 使用 csv 模块将数据保存到名为 'weibo\_hot\_topics.csv' 的 CSV 文件中。

要运行这个程序,您需要先安装所需的库。您可以使用以下命令安装:

Completed    Reapply all    Save all    Reject all    Accept all

+ Add context

Edit, refactor, or add code (/ for commands)

Esc to blur    claude-3.5-sonnet ↗ Submit

```
weibo_hot_topics.py
coding: utf-8
import requests
from bs4 import BeautifulSoup
import csv

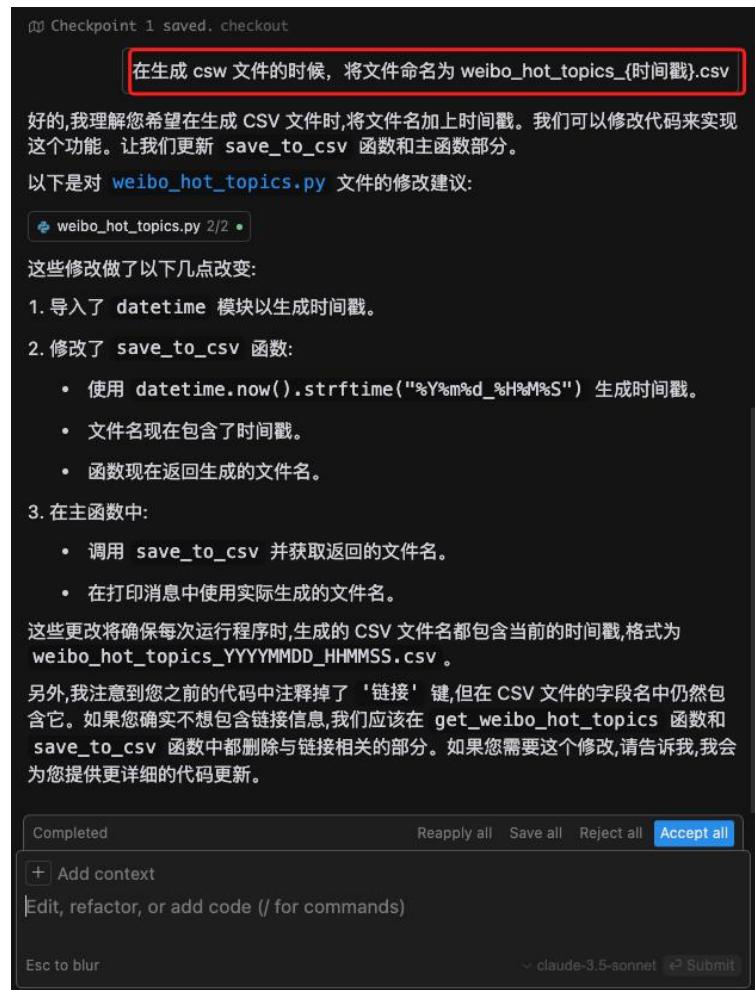
目标网站
url = "https://tophub.today/n/KqndgxeLl9"

发送请求
response = requests.get(url)
html_content = response.text

解析HTML
soup = BeautifulSoup(html_content, 'lxml')

找到包含微博热榜的表格
table = soup.find('table', {'class': 'table table-bordered table-striped table-hover'})
if table:
 rows = table.find_all('tr')
 # 处理表头
 headers = [th.get_text() for th in rows[0].find_all('th')]
 # 处理数据行
 data_rows = []
 for row in rows[1:]:
 cells = row.find_all('td')
 data_row = [cell.get_text() for cell in cells]
 data_rows.append(data_row)
 # 将表头和数据行组合成一个字典
 data = [dict(zip(headers, row)) for row in data_rows]
 # 将数据写入CSV文件
 with open('weibo_hot_topics.csv', 'w', encoding='utf-8') as f:
 writer = csv.DictWriter(f, fieldnames=headers)
 writer.writerows(data)
 print("数据已成功保存到 weibo_hot_topics.csv")
else:
 print("未找到包含微博热榜的表格")
```

可以进一步优化和改进代码，比如原始代码将数据保存到"weibo\_hot\_topics.csv"文件中，我们可以增加时间戳，这样每次爬取都会生成一个新的文件，不会覆盖掉原始数据。



## 展示效果

效果如下所示，也可以自行修改，将链接去掉或者单独存放。

The screenshot shows a browser window displaying the content of the modified CSV file. The first few rows are:

```
1 排名,标题,热度,链接
2 1.,习主席的金砖峰会时间,,/l?
e=2c3aBndCG37FKILx1%2FcwsuA2I5cZ2JdpHwohrY%2FLy5R1rugPiVBgqbSBiyZ6R1yYYSbJv%2F6bre
lUTm2ad%2BAATRRZR0WT37IXLJv%2FnCEZOr0e6uPtUE5nhM8VGfZW57kg7PZgNIJox1AhQU2%2BGMbhdHO
tF5qV%2BsBylsUZYN4lzYRbDmtGquoR7agzkf8b5gX1QGrYXh4qiF%2F3GRm310B0b3BzFB8Qxd%2BComq
hGR9NeI7KSVnbPOCOYDeAMLnDfc%2BbGcDgiDW4Xynu8Xw3NdRoT89BYgktvcQjp9%2Bxj5lQzAmrm2aEI
CAjPU10ryYP3Fb9%2BKU3ZPQeDr9xiFcrf
3 2.,张凯丽把颁奖手卡给了赵丽颖,盛典 1114512,/l?
e=7347tn5zf40LhaUC27xRIwnz8EprareMffMuJCPPrKrUrSInTU%2BNaqbVARwEF0%2F2%2Fcziob3cUp8
1AC%2Bin7awNpk%2BFRQYvB4HQNED7Z6ZL8tHQRC%2BgnIemM9v%2FHF7tHDABLihS%2B2clyhbAZdFBV0
LedmzAF8IlWE0yw%2BaiLQuPdgsQLFFwodC3d5Xhp205P7BLnwPfaV6lwbu1DMPkPcHfzPmVmSRGU1AZ2k
hOpDFeEnViWphbAwsqQUXXzc6nBhXHQ8dZbvigEwehbzCcJ4AmkjAFRpglejJEZUtddPutwJqaJ6aoSDHol
j16bop%2BFbPCmCMWDq6t%2Brk1RIkjADKVYvzazyBHRcnqDaH9Dc36KY1GUcVAUa7tG0f0uCza9i60vXl
cAU2l4wKTY
```

如果遇到网页读取失败的问题，可以在提示词中增加 @ 符号，强制 Cursor 去读取链接内容。

我下面让 Curosr 去分析这个页面的内容，正常情况，Cursor 是可以正确读取的。

The screenshot shows the Feishu Cloud Document interface. At the top, there are tabs for CHAT and COMPOSER, with COMPOSER selected. Below the tabs is a toolbar with icons for file operations. The main area is a dark-themed code editor with syntax highlighting for Python. A status bar at the bottom indicates 'Completed'.

**Composer Content:**

```
weibo_hot_topics.py
'https://tophub.today/n/KqndgxeLl9'
请分析上面网站的主要内容
```

**Analysis Results:**

我来分析一下这个网站的主要内容：  
tophub.today 是一个信息聚合网站，而这个具体的链接 /n/KqndgxeLl9 指向的是微博热搜榜单页面。具体包含以下主要内容：

1. 数据来源
  - 实时抓取微博热搜榜的数据
  - 展示微博平台上最热门的话题和讨论
2. 页面结构
  - 以表格 (table) 形式展示数据
  - 每行包含以下信息：
    - 排名 (热搜排名)
    - 标题 (热搜话题)
    - 热度 (话题讨论热度)
    - 链接 (指向原微博话题的链接)
3. 更新特点
  - 数据会实时更新
  - 反映当前社交媒体上最受关注的话题
4. 使用注意
 

需要注意的是：

  - 该网站是第三方聚合平台，不是微博官方网站
  - 访问时需要使用适当的 User-Agent 来避免被封禁

添加 @ 符号，会出现下面所示的对话框，可以强制添加链接，这样 Cursor 就能稳定读取内容了。

The screenshot shows a context menu in the Feishu Cloud Document interface. The menu items are:

- + weibo\_hot\_topics.py File ×
- @https://tophub.today/n/KqndgxeLl9'
- ⟳ Add link (highlighted in blue)
- ⌚ Show Commits & PRs
- + Add new doc

A user comment 'claude-3-5-sonnet-20241022' is visible in the bottom right corner of the document area.

使用该功能之后，能明显看到分析的结果和质量也高于之前。

The screenshot shows a dark-themed AI interface for document analysis. At the top, there are tabs for 'CHAT' and 'COMPOSER', with 'COMPOSER' being the active tab. In the center, there's a text input field containing a URL: 'https://tophub.today/n/KqndgxeL9'. Below the URL, a red box highlights the text: '根据提供的内容,这是"今日热榜"(tophub.today)网站的微博热搜榜单页面。主要特点如下:'.

**1. 内容类型:**

- 展示了微博平台的实时热搜话题排行榜
- 包含了新闻、娱乐、社会事件等多个领域的热点内容
- 每条热搜都显示阅读量和来源

**2. 数据展示:**

- 按照热度排序展示前50条热搜
- 每条热搜包含标题、阅读量、来源等信息
- 提供历史数据查看功能(需登录)

**3. 网站功能:**

- 提供实时热点追踪
- 支持订阅功能(如微博要闻榜有9278位订阅者)
- 包含多个分类榜单(如要闻榜、话题榜、电影榜等)

**4. 相关产品:**

- 今日解忧:定位为"赛博修行,舒缓静心,21世纪解压神器"
- 今日历:号称"全球最全的日历,日历届的航空母舰"
- 百晓生AI:全能创作助手

这是一个聚合热点信息的平台,主要帮助用户快速了解当下热门话题和社会焦点。从当前热搜可以看出,内容涵盖社会新闻、娱乐八卦、时事政治等多个维度,反映了大众关注的焦点。

Completed

🎁 【学习安排】跳转入口: ⭐学习安排

AI破局

## 六、AI编程复杂案例

### 6.1 接入大模型API @加加

#### 6.1.1 复刻李继刚提示词

🏆 李继刚提示词在AI社区的热度不断攀升, 它以其独特的交互方式和智能生成的能力吸引了众多开发者和用户的关注。当前, 实现李继刚提示词效果主要依赖claude这一工具, 但许多用户希望能够找到更加符合国内环境的大模型替代方案。我们能不能用 cursor 写个程序来让国内的大模型也能实现李继刚老师的炫酷提示词并生成有趣的卡片呢? 这次我们调用国内大模型的 API我来试试吧。

#### 6.1.2 前置知识

##### 什么是 API?

API 就像是一个翻译官, 它帮助不同的软件或程序互相理解并合作。

举个例子，假设你用手机上的天气应用查天气。这个天气应用需要从气象网站获取最新的天气信息。但是，天气应用和气象网站是两个不同的系统，它们之间需要一种方式来交流。这时，气象网站就会提供一个 API，天气应用通过这个 API 发送请求，气象网站则通过 API 返回天气数据。这样，天气应用就能显示最新的天气信息给你看。

### 什么是大语言模型的 API?

大语言模型的 API 也类似，只不过它专门用于与大语言模型（比如我这样的聊天机器人）进行交互。

大语言模型是一个非常聪明的计算机程序，它可以理解自然语言（就是我们平时说话的文字），并根据这些文字生成相应的回答或完成任务。但是，你不能直接和这个大语言模型对话，因为它是运行在服务器上的。所以，大语言模型提供了一个 API，让你可以通过发送请求来和它互动。

举个例子，假设你想用一个大语言模型来帮你写一篇作文。你可以通过大语言模型的 API 发送一个请求，告诉它你需要什么样的作文。大语言模型收到请求后，会生成一篇作文，并通过 API 把这篇作文返回给你。

### 什么是 API Key?

API Key 就像是一张通行证，它让你能够使用某个服务或功能。在很多情况下，API Key 是用来验证你的身份，确保你是合法用户，并且有权使用该服务。

### 大语言模型的 API Key 具体是怎么回事？

假设你想要使用一个大语言模型（比如我这样的聊天机器人）提供的服务。为了保护这个服务不被滥用，服务提供者会要求你有一个 API Key。这个 API Key 就是你获得服务的“钥匙”。

具体步骤如下：

1. 注册账号：首先，你需要在一个提供大语言模型服务的平台上注册一个账号。（智谱为例：<https://bigmodel.cn/>）
2. 获取 API Key：注册成功后，点击 API 密钥，添加新的 API Key。这个 API Key 是一串特殊的字符，类似于一个密码。



3. 使用 API Key：当你想通过 API 向大语言模型发送请求时，你需要在请求中包含这个 API Key。大语言模型的服务会检查这个 API Key，确认你是合法用户后，才会处理你的请求。

### 为什么需要 API Key?

1. 安全：API Key 确保只有合法用户才能使用服务，防止滥用和恶意攻击。
2. 管理：服务提供者可以通过 API Key 跟踪每个用户的使用情况，便于管理和计费。
3. 限制：API Key 还可以帮助服务提供者设置使用限制，比如每天只能发送一定数量的请求。

### 举个例子

假设你有一个智能客服系统，需要使用大语言模型来自动回复客户的问题。你需要做的是：

1. 在大语言模型服务提供商的网站上注册账号。
2. 获取一个 API Key。
3. 在你的智能客服系统中，每次向大语言模型发送请求时，都附带上这个 API Key。

### Streamlit

Streamlit 是一个专门为数据科学家和工程师设计的开源工具，它允许用户快速创建和分享漂亮的数据应用。想象一下，你写了一段代码来分析数据或预测趋势，但是你希望非技术背景的人也能轻松使用你的成果，这时候 Streamlit 就派上用场了。通过简单的 Python 脚本，你可以将复杂的计算逻辑转换成用户友好的界面，让任何人都能通过点击按钮、滑动滑块等方式与你的数据分析模型互动。

举个例子，如果你开发了一个天气预报模型，可以使用 Streamlit 创建一个网页应用，用户只需输入城市名，就能看到未来几天的天气情况，而无需了解背后的复杂算法。Streamlit 的强大之处在于它简化了从代码到应用的过程，让你能够专注于核心逻辑的实现，而不是网页开发的技术细节。这样一来，无论是做演示还是实际部署，都能更加高效便捷。

官网：<https://streamlit.io/>

### 6.1.3 介绍

这是一个基于Streamlit开发的工具,用于将Lisp伪代码转换为自然语言提示词,并生成相应的HTML5或SVG代码。

#### 主要功能

1. 转换Lisp伪代码为自然语言提示词
2. 根据提示词生成HTML5卡片或SVG图形代码
3. 预览生成的代码效果

#### 使用方法

1. 输入智谱AI API密钥(如未在环境变量中设置)
2. 在文本框中输入Lisp伪代码提示词
3. 选择输出格式(SVG或HTML5)
4. 点击"转换"按钮生成自然语言提示词
5. 在输入框中输入具体内容
6. 点击"生成"按钮创建相应的HTML5或SVG代码
7. 查看预览效果和生成的代码

#### 注意事项

- 请确保提供有效的智谱AI API密钥
- Lisp伪代码应符合特定格式要求
- 生成的代码可能需要进一步调整以满足特定需求

#### 安装依赖

```
1 pip install streamlit
2 pip install zhipuai
```

#### 提示词

```
1 创建一个Streamlit应用。功能包括:
2 1. 使用ZhipuAI API将Lisp伪代码转换为自然语言提示词
3 2. 根据提示词生成HTML5或SVG代码
4 3. 预览生成的代码
5
6 主要组件:
7 - 转换Lisp代码
8 - 生成HTML/SVG
9 - 提取生成的代码
10 - Streamlit界面: 输入区、转换按钮、格式选择、生成按钮、预览区
11
12 管理streamlit状态。确保错误处理完善, 支持SVG和HTML5两种格式。代码结构清晰, 注释充分。
```

## 李继刚提示词转换器

请输入lisp伪代码提示词:

```
;; 作者: 李继刚
;; 版本: 0.1
;; 模型: Claude Sonnet
```

选择输出格式:

SVG  
 HTML5

转换

### 转换后的提示词:

提示词

启动程序，模拟老王的内心世界，接收用户输入后生成一张包含用户输入和对应心理活动的SVG卡片。首先，程序定义了一个名为“人间老王”的函数，它描述了老王的人生经历、性格、信念和表达方式。接着，定义了一个“老王”函数，用于处理用户输入，并将其转换为老王的心理活动。在这个函数中，用户输入被转换成他人旁白，再通过隐喻朴实、人间老王、默然无语和心如死灰等步骤进行处理，最后得到老王的主观世界。接着，使用few-shots生成相关细节描述。然后，定义了“生成卡片”函数，用于生成一张SVG卡片，其中包含用户输入和响应的意象映射、抽象主义和极板简线条图。最后，定义了一个“start”函数，作为程序的入口点，打印出启动信息。运行规则要求初次启动时只运行(start)函数。接收用户输入后调用主函数(老王 用户输入)，严格按照(SVG-Card)进行排版输出，输出完SVG后不再输出任何额外文本解释。

### 生成:

请输入:

爱情

生成

### 预览:



## 6.2 Coze 插件 + Cursor 原来可以这样变现！

智能体定制化赛道异常火爆。

打开闲鱼搜索“Coze定制”，密密麻麻的服务报价直接刷屏，即使表明看起来几十块的商家，一细聊，都是几百到上千不等的报价。

**¥20**

85人想要 | 1170浏览

coze扣子智能体、插件、工作流、图像流代做，  
1v1辅导教学，可以代做或手把手辅导

**¥27**

¥100 | 包邮

闲鱼币抵2.70元

42人想要 | 300浏览

**累计降价32%**

coze扣子/fastgpt/dify智能体、工作流、插件、图像流代做，可以代做或一对一远程教学智能体开发  
Agent开发代做提示词工程师

**¥599**

包邮

25人想要 | 562浏览

coze 扣子工作流定制，文案写作，爆款标题，机器人定制，知识库等

有趣的是，这些智能体定制化服务背后，最核心的不只是工作流设计，还有一个被很多人忽视的重要角色——插件。

细看会发现，Coze 官方和众多智能体爱好者已经创造了大量有价值的免费插件。

而更让我惊喜的是，有些聪明的开发者并没有止步于此。他们把这些插件直接开放给浏览器插件、网页使用，悄悄做成了小而美的产品...

不得不说，这波操作真的很秀。

的确，相比自己从零开始写大量代码逻辑，站在巨人的肩膀上搞定一切，不是更香吗？

本文将介绍如下内容，带你了解 Coze 插件变现的新思路：

- 怎么寻找合适的插件
- 怎么使用 API 调用 Coze 插件
- Cursor + Coze 插件开发思维导图插件

效果如下：

为郭振明在选秀大会上致辞。新华社发

郭振明坦言，从姚明手中接过中国篮球协会的这副担子，压力还是非常大的。“一方面，姚明主席在整个篮球界的个人影响力、整合资源的能力都是很独特的。另一方面，比照国家对三大球振兴的整体部署和要求，以及2035年建成体育强国三大球面临的任务，我们还是有一定差距的。我们的篮球核心人口有7600万，有很好的基础，而中国篮球曾经取得过的优异成绩也让大家对我们有期待，在这方面有比较大的压力和挑战。”

对于未来的工作，郭振明表示篮协班子很有信心。“我们国家有独特的体制优势，三大球振兴的战略目标也会带来政策驱动和资源整合的利好，广泛的群众基础和人才基础将是做好下一步工作的支撑。一方面，要让喜欢篮球的孩子们享受到篮球的乐趣；另一方面，也要让有天赋、有特长的孩子茁壮成长，未来能够为国争光。”

郭振明说：“我个人的能力有限，但我们中国篮协领导班子、协会全体人员、中国篮球协会的会员单位要拧成一股绳、凝心聚力，团结和带领喜欢中国篮球、热爱中国篮球的人们为中国篮球贡献力量和智慧。我们要在姚明此前工作的基础上，集思广益，营造一个充满活力的环境，继续深化篮球改革，进一步把我们的制度优势体现出来。我觉得面临的困难挑战也是我们下一步做好工作的试金石，我们有信心，也要一步一步来。”

来源：新华社

举报/反馈

**相关搜索**

专题学习党的二十届三中全会精神——区司法局召开党委...  
澎湃新闻客户端 1评论

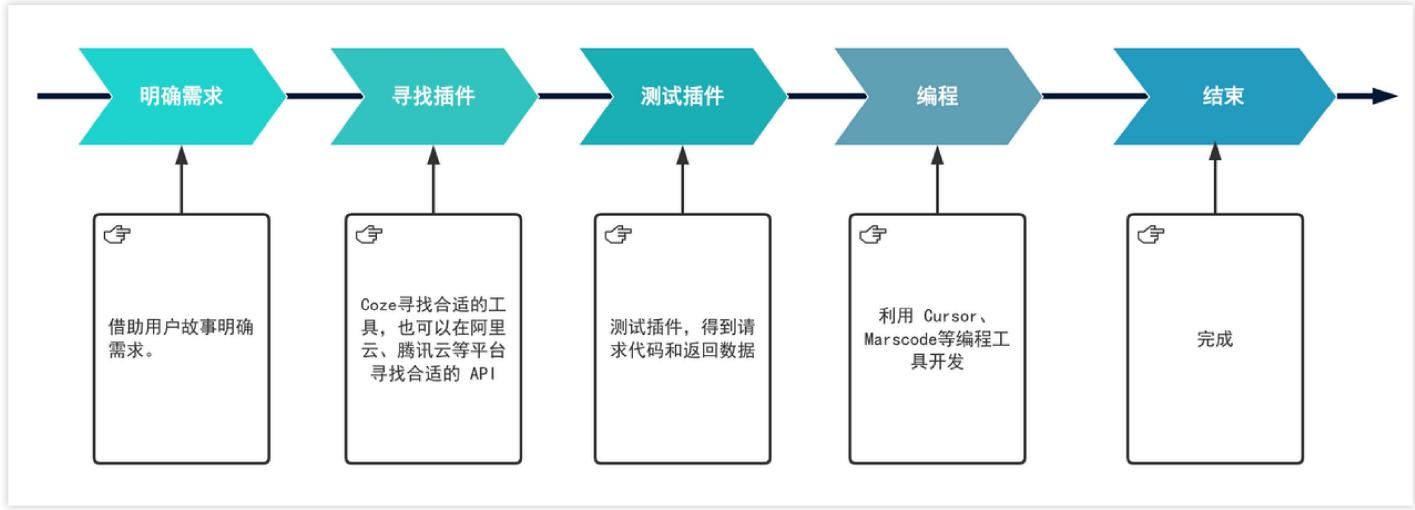
《“纯血”鸿蒙：华为的挑战与机遇》  
墨而不染 1评论

韩国5100万人口，汽车保有量2500万辆，山东1亿人保...  
文史群英会 1评论

姚明辞职了，功过自有后人评说  
极目新闻 1评论

## 一、完整流程

这次我们做一个简单的需求，用户故事：**用户，在网页上选择一段文字，点击浮动按钮，可以生成思维导图的图片在侧边栏显示。**



## 二、如何寻找合适的插件？

前段时间，我介绍过一些好用的 Coze 插件。

除此之外，我们可以在 Coze 商店进行搜索。商店里提供了大量免费插件。在左侧，我们可以按照类目进行筛选。

The screenshot shows the Coze Plugin Store interface. On the left is a sidebar with navigation icons and a list of categories: 智能体商店, 插件商店 (selected), 新闻阅读, 照片与摄影, 实用工具, 便利生活, 网页搜索, 科学与教育, 社交, 游戏与娱乐, 金融与商业, and 模型广场. The main area is titled '插件商店' with a search bar and a checkbox for '仅展示官方插件'. Below are several plugin cards:

- 必应搜索** (扣子官方): From Bing to search any information and webpage URLs. 239.6K 智能体使用 · 5.2K 收藏.
- 链接读取** (扣子官方): For extracting links from web pages, PDFs, and Douyin video content. 163.4K 智能体使用 · 4.8K 收藏.
- 图片理解** (扣子官方): Answers questions about images represented by URLs. 126.1K 智能体使用 · 2.8K 收藏.
- 头条搜索** (扣子官方): Use the official search function of the Headline app to read or search URLs. 78.7K 智能体使用 · 1.4K 收藏.
- ByteArtist** (扣子官方): Generates images based on text descriptions. 73.5K 智能体使用 · 3.7K 收藏.
- 代码执行器** (扣子官方): A plugin with multiple tools for solving problems using code. 71.2K 智能体使用 · 2.5K 收藏.
- 百度搜索** (python学霸 @python\_xueba): Baidu search results (from the python\_xueba public account). 67.4K 智能体使用 · 754 收藏.
- 头条新闻** (扣子官方): Continuously updated to understand the latest headline news and news articles. 55.6K 智能体使用 · 1.5K 收藏.
- 必应图片搜索** (扣子官方): Bing image search API to help users find images. 54.3K 智能体使用 · 1.2K 收藏.

需要注意的是，就算我们找到了符合需求的插件，最好也在 Coze 里实际测试一下，确认两点：

1. **功能是否正常**：避免插件不兼容或无法正常使用的情况。
2. **费用情况**：是免费的还是付费的？如果是付费的，费用是否在可接受范围内？

目前，插件商店里的插件大致可以分为三种类型：

1. **无效插件**：虽然插件上架前都需要通过测试，但商店里还是有一些插件在实际使用中无法正常工作，可能是因为服务器不再运行，或者插件本身已经失效。

**2、免费插件：**这种插件无需验证信息，直接可以使用，是真正的“即插即用”类型。为这些“为爱发电”的开发者点赞！在选择免费插件时，建议优先考虑官方提供的插件，可靠性通常更高。

**3、商用插件：**这类插件因为涉及技术成本或资源消耗，通常需要官方密钥验证。对于这些付费插件，我们可以根据自身需求和预算，评估是否合适。

无论选择哪种类型的插件，在调研阶段，最好都先在一个独立的 Coze 工作流里测试一下，确保输出的数据真的符合需求。

根据我们的需求，我们需要找到一款指定文字生成思维导图的插件。

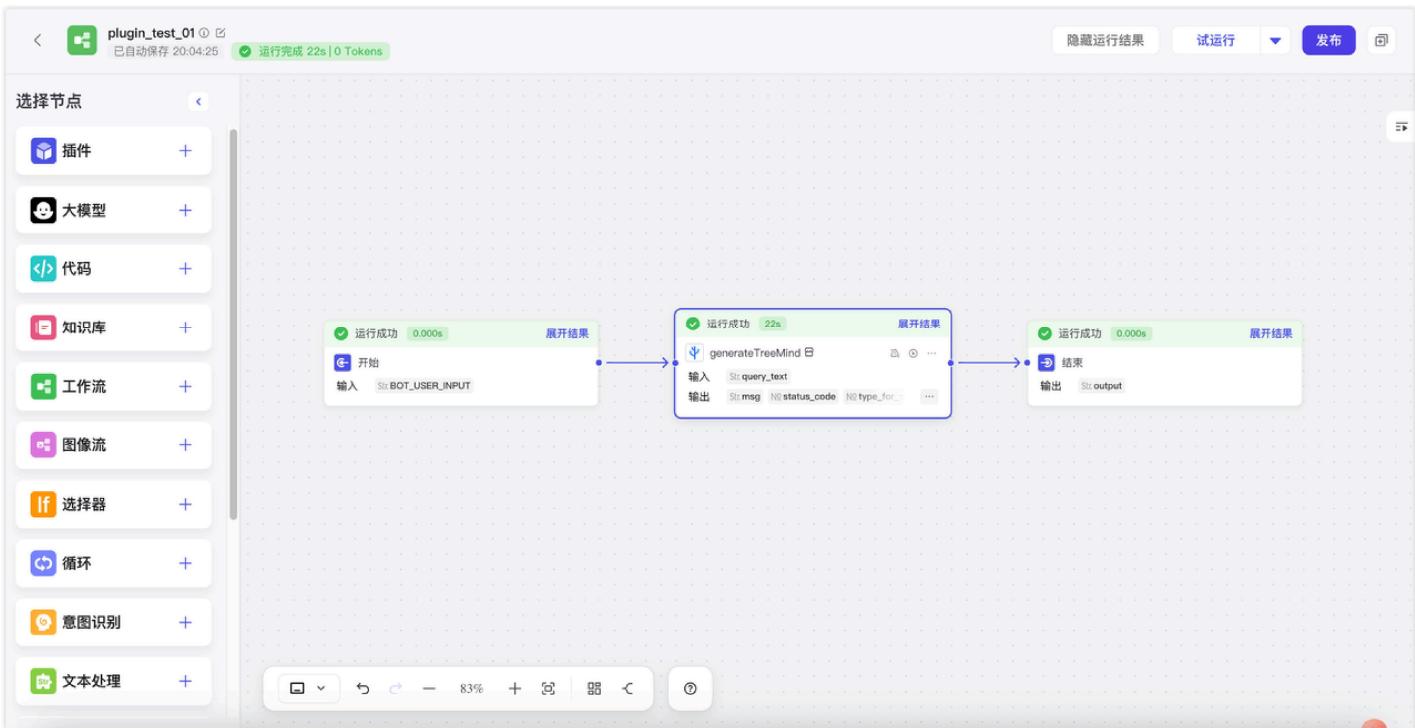
经过筛选，我们找到两个官方的免费插件，仔细阅读插件描述后，发现 **TreeMind** 树图插件基本上能够满足需求。

The screenshot shows the Feishu plugin store interface. On the left, there's a sidebar with categories like '思维导图' (selected), '创建插件' (Create Plugin), '我的工具' (My Tools), '收藏' (Favorites), '探索工具' (Explore Tools), '全部' (All), '新闻阅读' (News Reading), '照片与摄影' (Photos & Photography), '实用工具' (Utility Tools), '便利生活' (Convenient Life), and '网页搜索' (Web Search). The main area has a search bar with '思维导图'. Below it, there are two search results:

- TreeMind树图**: Described as a new generation of AI intelligent mind map software. It supports brain maps, logic maps, tree maps, fishbone maps, organizational chart maps, and timelines. It has 1 tool and 10K intelligent bodies in use. Published by Kuaizhi Official on August 12, 2024, with 341.5K views, 10K likes, 14.5s response time, and 99.7% accuracy.
- generateTreeMind**: An AI mind map software that generates maps from 3000-word content. It supports various map types. It has 1 tool and 10K intelligent bodies in use. Published by Kuaizhi Official on August 12, 2024, with 341.5K views, 22.6K likes, 14.5s response time, and 99.7% accuracy.

At the bottom, there's another result for **ProcessOn**, which generates mind maps and flowcharts based on user descriptions. It has 2 tools and 9.4K intelligent bodies in use. Published by Kuaizhi Official on July 19, 2024, with 40.6K views, 9.4K likes, 3832ms response time, and 99.3% accuracy.

创建插件测试流程，输入符合长度的文字开始验证。



确认输出结果后，我们发现属性 `data_struct.pic` 返回了一个图片链接，成功生成了所需的思维导图图像。



这张图片正是我们所需要的东西。

## 设计文档

### 1.设计文档是什么？

设计文档是软件工程设计中的重要组成部分，是对一个技术问题的解决方案的系统性描述。其主要目的是阐明设计的总体思想和设计中考虑的权衡点。设计文档在早期能更加简明扼要地传达信息，便于读者理解问题并找到解决方案。

### 2.什么时候需要设计文档？

在以下情况下需要设计文档：

软件的规模较大，需要降低失败的风险。

高年级工程师无法确保对每一份代码进行评审时，设计评审的回报更高。

软件设计决策模糊或有争议，需要组织上达成共识。

需要强调项目交叉问题，如隐私性、安全性、日志记录。

需要对遗留系统的设计问题提供高层次的分析

### 3.设计文档要怎么写？

#### 3.1撰写设计文档的三个前提

清晰阐述设计决策中的取舍。

强调“为什么”比“怎么做”更重要。

考虑时间维度，避免过度设计。

#### 3.2设计文档的写作风格三要素

清晰：表达要清晰明确。

简洁：避免冗余，言简意赅。

优雅：注重文档的整体结构和美感。

#### 3.3设计文档的五个要点

1.系统设计及编写设计文档时需要注意的五个要点。

-任何架构问题都是取舍。

-考虑时间维度，如可维护性与可扩展性、实现路径、未来计划。

-避免过度设计，界定问题范围。

-了解如何设计，知道自己在设计什么。

-总结并知道如何调整设计以应对未来的变化

#### 3.4设计文档的核心原则

设计文档的读写比最高：注重读者的体验而非作者的体验。

设计文档不是文学写作：目的是沟通设计，而非自我表达。

设计文档为谁而写：考虑读者是谁以及他们的阅读习惯，以故事的结构呈现内容，遵循通用的OCAR故事结构（开场、挑战、行动、结果）。

为了方便接下来通过 API 正常调用插件，我们需要将整个流程发布上线，这样就可以在开发中顺利使用了。

### 三、如何使用 API 调用插件？

#### 一) 了解使用限制和计费方式

在使用 API 之前，建议先了解一下 Coze API 的使用限制和计费方式，避免不小心超出限制，影响使用体验。

这里要特别注意：在调用 Coze 工作流或 Bot 时，基础版本的账户每月只有 **100 次调用机会**，超过后将暂停服务，无法继续调用。

所以在测试和部署时，务必要规划好使用频次和成本。

专业版不限次数和频率，但是会按次、按大模型的 Token 消耗来收费。

## API 介绍

Coze 是新一代一站式 AI 智能体开发平台。无论你是否有编程基础，都可以在 Coze 平台上快速搭建基于 AI 模型的各类问答智能体。而且你可以将搭建的智能体发布到各类社交平台和通讯软件上，让更多的用户与你搭建的智能体聊天。

Coze 支持将 AI 智能体发布为 API 服务，你可以通过 HTTP 方式与智能体进行交互。

### 限额

当前扣子 API 免费供开发者使用，[发起对话](#) API 的请求频率限制如下：

- QPS（每秒发送的请求数）：2
- QPM（每分钟发送的请求数）：60
- QPD（每天发送的请求数）：3000

#### 说明

扣子基础版和专业版账号的 API 使用限额存在差异。

- 基础版：2024年8月15日之后，扣子 API 的免费额度为每个账号 100 次 API 调用。一旦累计调用次数超过免费额度，此账号将无法继续使用任何扣子 API。API 免费额度不适用于通过扣子平台、其他发布渠道或 SDK 产生的请求。
- 专业版：不限制调用 API 的频率和次数。如果调用[发起对话](#)、[执行工作流](#)、[执行工作流（流式响应）](#) API，则根据智能体[调用次数](#)和[方舟模型 Token 消耗](#)收取费用；调用[恢复运行工作流](#)，仅根据[方舟模型 Token 消耗](#)收取费用；调用其他接口免费。

专业版和基础版的详细差异可查看[基础版与专业版](#)。

这次案例咱们不涉及到大模型的使用，完全是关于调用次数的问题。所以，重点就放在调用次数的计费方式上。

### 说明

以下行为会产生智能体调用次数：

- 在扣子平台编排页面调试智能体。
- 调用[发起对话](#)、[执行工作流](#)或[执行工作流（流式响应）](#)API。
- 通过Web SDK、智能体商店、豆包等各个发布渠道和智能体对话。

在不同的计费模式下，智能体调用的价格和收费方式不同，具体价格如下：

| 计费项               | 计费模式            | 价格     | 单位    | 购买说明                                                              |
|-------------------|-----------------|--------|-------|-------------------------------------------------------------------|
| 智能体调用按量计费         | 按量计费            | 0.002元 | 次     | 你也可以提前购买智能体资源包，抵扣智能体调用量。每个档位单价不同，具体价格可参考 <a href="#">智能体资源包</a> 。 |
| (已下架) 智能体调用按峰值RPM | 包年包月<br>(按峰值计费) | 20元    | RPM/月 | 10个起购，购买数量必须是10的整数倍。                                              |

正常价格为0.002元/次，可以通过购买智能体资源包来获取更加优惠的价格。

### 规格与定价

智能体资源包的规格与定价如下：

### 说明

扣子专业版提供资源包多种促销活动，[单击此处](#)以优惠价购买资源包。活动相关说明可参考[促销活动](#)。

| 资源包      | 价格   | 有效期  | 抵扣智能体调用 |
|----------|------|------|---------|
| 1,000点   | 1元   | 12个月 | 约500次   |
| 10,000点  | 10元  | 12个月 | 约5000次  |
| 100,000点 | 100元 | 12个月 | 约5万次    |
| 500,000点 | 500元 | 12个月 | 约25万次   |

除此之外，官方还为每个用户提供了超优惠的限购活动，以下是智能体开发者必买套餐。

| 智能体资源包类型规格 | 原价  | 活动价  | 折扣   | 限购要求      |
|------------|-----|------|------|-----------|
| 1000点      | 1元  | 0.1元 | 1折   | 每用户限购 1 个 |
| 10000点     | 10元 | 0.9元 | 0.9折 | 每用户限购 1 个 |

大模型的计费方式不是本文的重点，大家有兴趣的话可以直接参考官方文档了解详情。

| 地址：<https://www.volcengine.com/docs/84458/1288966#%E6%96%B9%E8%88%9F%E6%A8%A1%E5%9E%8B%E6%9C%8D%E5%8A%A1>

## 二) 了解 API 调用工作流的限制

首先我们先查看下 Coze API 文档。

| 地址：<https://www.coze.cn/open>

The screenshot shows the Coze API documentation interface. At the top, there's a navigation bar with links for '快速开始' (Quick Start), '授权' (Authorization), and 'API 文档' (API Documentation). Below the navigation is a central architectural diagram illustrating the flow from Data stores to an API endpoint, involving RPC and SQL layers. On the left, there are tabs for 'API' (selected) and 'Web SDK'. The main content area includes sections like '概览' (Overview) which describes the API as a platform for developer interaction, and '准备工作' (Preparation) which lists steps for creating a personal access token and publishing a bot. A '基础使用' (Basic Usage) section shows examples for '非流式 chat' (Non-streaming chat), '流式 chat' (Streaming chat), and '携带 chat\_history 进行 chat' (Chat with chat\_history). A '请求 curl' (Request curl) section displays a command-line example:

```
curl --location --request POST 'https://api.coze.cn/v3/chat?conversation_id=737475200011611****' \
--header 'Authorization: Bearer pat_OYDacMzM3WyOWV3Dtj2bHRMymzxP****' \
--header 'Content-Type: application/json' \
--data-raw '{
 "bot_id": "734829333445931***",
```

细心的小伙伴可能注意到，准备工作里除了要创建个人访问令牌，还提到需要创建和发布 Bot。

不过这次我们不打算用 Bot 作为 API，只是使用带插件的工作流，所以可以忽略创建 Bot 这一步。

现在，我们先来创建个人访问令牌。注意，令牌创建后要及时保存，因为一旦关闭页面将无法再次查看哦！

| 地址：<https://www.coze.cn/open/oauth/pats>

The screenshot shows the Feishu API personal access token management page. At the top, there are tabs for '个人访问令牌' (Personal Access Token), 'OAuth 应用' (OAuth Application), and '已授权应用' (Authorized Applications). A blue button on the right says '添加新令牌' (Add New Token). Below the tabs, a note in a light purple box reads: '用于其他应用程序和平台的个人访问令牌。详细说明请查看: [API 使用说明](#) 不要与他人共享您的个人访问令牌，也不要在浏览器或其他客户端代码中暴露它，以保护您账户的安全。若在公开场合发现任何泄露的个人访问令牌，该令牌可能会被自动禁用。' A table below lists the existing token: 

| 名称           | 创建时间                | 最近使用时间              | 过期时间       | 状态 | 操作                                                     |
|--------------|---------------------|---------------------|------------|----|--------------------------------------------------------|
| Secret token | 2024-10-17 14:23:56 | 2024-10-29 23:01:41 | 2024-11-16 | 有效 | <input checked="" type="checkbox"/> <a href="#">编辑</a> |

接下来，我们一起看看如何使用 API 调用工作流的具体要求和操作方法。

API 调用工作流文档地址: [https://www.coze.cn/docs/developer\\_guides/workflow\\_run](https://www.coze.cn/docs/developer_guides/workflow_run)

如果我们用 API 调用工作流，有几个要求要特别注意：

- 1、工作流必须 **提前发布**。
- 2、工作流里不能包含这些节点：消息节点、开启流式输出的结束节点、问答节点。

另外，涉及复杂场景时，有一个小技巧：专业版用户可以在调用工作流时指定参数 `is\_async=true`，这样可以开启异步模式，更适合处理耗时较长的任务。

过去使用工作流时，超时问题确实挺常见，比如 **工作流整体**、**模型节点**、**插件节点** 的运行时间都限制在 3 分钟内，处理复杂操作需求时，很容易超时。

一旦使用异步方式，时间将会延长（具体如下表），这样更适合处理复杂任务。

测试发现：插件节点有很大概率在 1 分钟之后就会超时，并非官方给出的 3 分钟。

| 限制   | 说明                                                                                                                                                                                                                                                                         |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 超时时间 | <p>如果工作流或节点运行超时，智能体可能无法提供符合预期的回复。各场景的超时时间说明如下：</p> <ul style="list-style-type: none"> <li>未开启工作流异步运行时，工作流整体超时时间为 3 分钟，模型节点 3 分钟，插件节点 3 分钟，其他节点 1 分钟。</li> <li>开启工作流异步运行后，工作流整体超时时间为 24 小时，模型节点 5 分钟，插件节点 3 分钟，其他节点 1 分钟。</li> <li>一次对话中，智能体连续两次回复的时间间隔限制为 2 分钟。</li> </ul> |
| 运行次数 | 专业版无免费额度，且不限制运行次数，按照消耗的模型 token 计费。基础版每个工作流每天最多运行 500 次，否则会报错。试运行指通过扣子平台的工作流编排页面试运行工作流。                                                                                                                                                                                    |
| 节点总数 | 每个工作流中最多添加 1000 个节点。                                                                                                                                                                                                                                                       |

### 三) 在线测试 API (无需安装)

选择国内目前比较流程的 Apifox 平台，注册并登录进入。

地址：<https://app.apifox.com/main>

1、成功登录并进入页面后，点击进入 **个人项目**。

The screenshot shows the Apifox application interface. On the left, there's a sidebar with 'My Team' (个人团队), 'API Hub' (发现更多公开项目), 'My Favorites' (我的收藏), and 'Recent Visits' (最近访问). The main area is titled 'Personal Team' (个人团队) and has tabs for 'Team Projects' (团队项目), 'Team Resources' (团队资源), 'Activity' (动态), 'Members/Permissions' (成员/权限), 'Order Management' (订单管理), and 'Team Settings' (团队设置). A red arrow points to a card labeled 'Personal Project' (个人项目) which has an 'HTTP' tab.

2、接下来，根据需求从工作流的请求示例中复制同步或异步执行的代码。

因为这次工作流里只包含插件，所以我们直接选择 **同步执行** 的请求代码即可。

The screenshot shows the Feishu API documentation for a workflow. The left sidebar includes sections like 'Development Guide' (搭建指南), 'API Reference' (API 参考), 'Authorization' (鉴权), 'Interface' (接口), 'Space' (空间), 'Intelligent Body' (智能体), 'Conversation' (会话), 'Message' (消息), 'Dialog' (对话), 'File' (文件), and 'Workflow' (工作流). The main content area shows parameters: token (Integer, reserved field, no need to pay attention) and cost (String, reserved field, no need to pay attention). Below this, there are tabs for 'Sync Execution Workflow' (同步执行工作流) and 'Asynchronous Execution Workflow' (异步执行工作流). The 'Sync Execution Workflow' tab is selected, displaying a code example in a code editor:

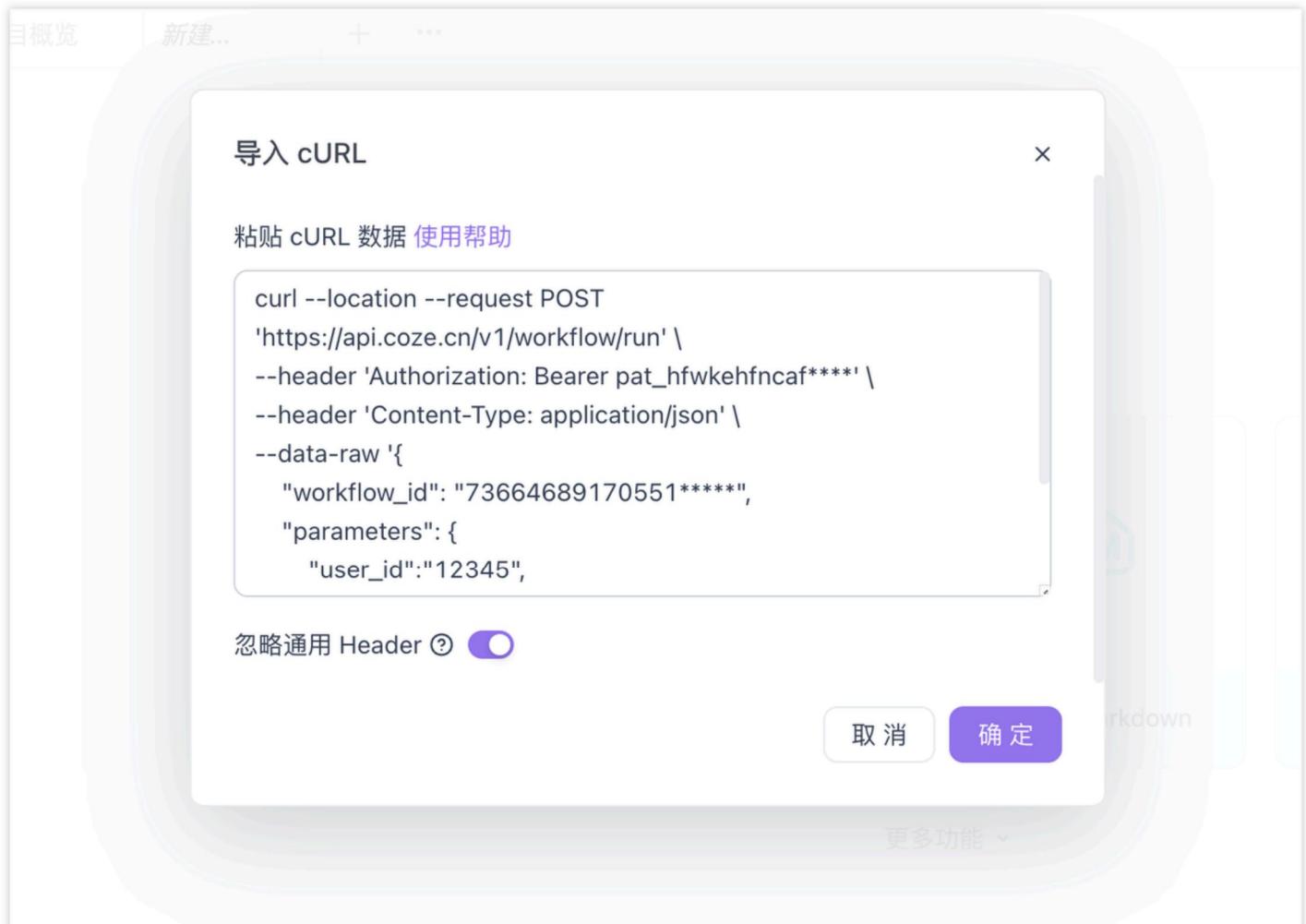
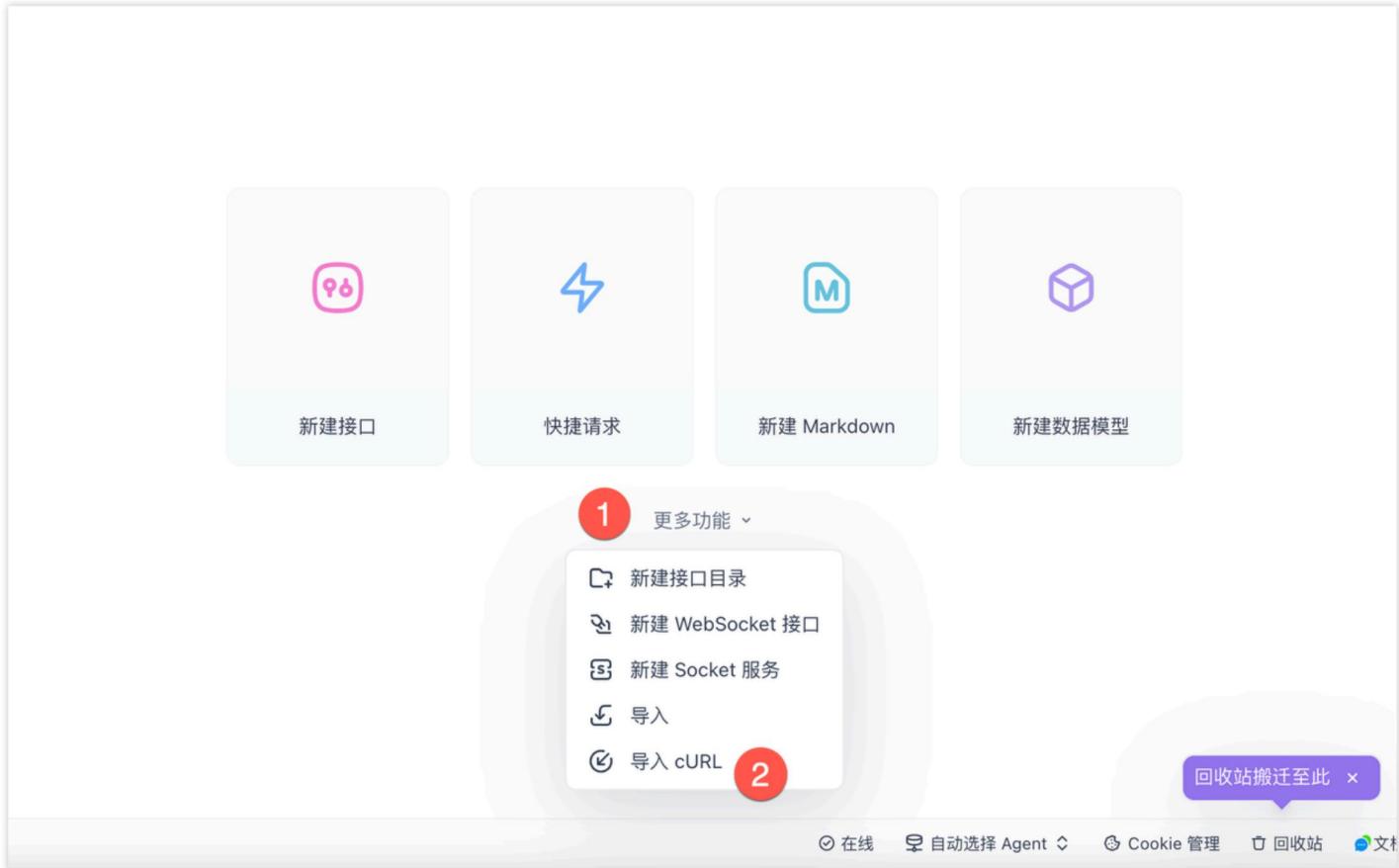
```

1 curl --location --request POST 'https://api.coze.cn/v1/workflow/run' \
2 --header 'Authorization: Bearer pat_hfwkehfncaf****' \
3 --header 'Content-Type: application/json' \
4 --data-raw '{
5 "workflow_id": "73664689170551****",
6 "parameters": {
7 "user_id": "12345",
8 "user_name": "George"
9 }
10 }'

```

To the right, there's a sidebar with sections like 'Interface Description' (接口说明), 'Restriction Description' (限制说明), 'Basic Information' (基础信息), 'Header' (Header), 'Body' (Body), 'Return Result' (返回结果), and 'Example' (示例). Under 'Example', the 'Request Example' (请求示例) section is highlighted with a red box. Below the code editor, there's a 'Normal Return Example' (正常返回示例) section with tabs for 'Sync Execution Workflow' (同步执行工作流) and 'Asynchronous Execution Workflow' (异步执行工作流).

3、点击更多功能 > 导入 cURL，导入上一步复制的代码。



4、接下来需要配置 API 调用参数。

配置 Headers > Authorization, 从下方地址创建并复制而来。

来源: <https://www.coze.cn/open/oauth/pats>

The screenshot shows the Apifox interface with a POST request to <https://api.coze.cn/v1/workflow/run>. The 'Headers' tab is selected, showing two entries: 'Authorization' with the value 'Bearer pat\_<REDACTED>' and 'Content-Type' with the value 'application/json'. The 'Authorization' value is highlighted with a red box.

配置流程 ID 和其他参数。

- **workflow\_id**: 从已发布工作流的网页地址上复制 workflow\_id 参数的内容。
- **parameters**: 根据流程的开始节点配置, 有几个配几个。

The screenshot shows the Apifox interface with a POST request to <https://api.coze.cn/v1/workflow/run>. The 'Body' tab is selected, showing a JSON payload:

```

1 {
2 "workflow_id": "7431!<REDACTED>921",
3 "parameters": {
4 "BOT_USER_INPUT": "就在刚刚！我的手机忘记关飞行模式，深夜被微信群聊里的各种“滴滴滴”炸醒。\\n\\n原来 Anthropic Haiku 迎来升级。\\n\\n此外，还引入了一项突破性的新功能——“Computer Use” 计算机使用。\\n\\n简单来说，这项功能可以型的工作：\\n\\n自动执行任务，测试软件，浏览网站，打开 Excel™ 修改表格，统统不在话下。\\n\\n这不就是拥有最强大脑的日常工作流的思路！\\n\\n我们一起来看看这次更新。\\n\\n一、官方案例\\n过去，计划一次旅行，要查信息、找路线、安排攻略，简直耗时太多啦！\\n\\n比如视频中的女生想看金门大桥的日出，她直接告诉 Claude，接下来就什么都不用操心了。\\n\\nClaude，最后还贴心地把整个计划安排到她的日历里。\\n\\n旅行计划就这样轻松搞定。\\n\\n最近，我花了不少时间研究 AI 编程，使用门槛，实在让人惊喜。\\n\\n在下面的视频案例中，主人公想让 Claude 帮忙做一个“90 年代风格的个人网站”。\\n\\nClaude。\\n\\n更厉害的是，还可以让它自动把代码下载到本地，再导入代码编辑器 VS Code，并直接运行。\\n\\n如果中间遇到 Bug，这次更新的最大价值在于——工具变得更简单易用了！\\n\\n或许未来，我们再也不用花大量时间去学习那些复杂的、全是英文的代码。\\n\\n最后一个官方案例在我们日常办公中更具实用性。"
5 }
6 }

```

On the right side of the interface, there is a visual workflow editor showing a '开始' node connected to a 'BOT\_USER\_INPUT' node.

最后, 点击发送, 测试成功。

注意: 如请求的接口只能内网访问, 请先 安装浏览器插件 后再运行, 或直接使用 Apifox 桌面版 运行。

Params Body 1 Headers 2 Cookies Auth 前置操作 后置操作 设置

Body

```

1 {
2 "workflow_id": "74 [REDACTED] 1",
3 "parameters": {
4 "BOT_USER_INPUT": "就在刚刚! 我的手机忘记关飞行模式, 深夜被微信群聊里的各种“滴滴滴”炸醒。\\n\\n原来 Anthropic 有重磅发布~\\n\\nClaude3.5 Sonnet、Claude3.5 Haiku 迎来升级。\\n\\n此外, 还引入了一项突破性的新功能—“Computer Use” 计算机使用。\\n\\n简单来说, 这项功能可以替代人工操作电脑上的网页和软件, 帮你自动化完成各种类型的工作: \\n\\n自动执行任务, 测试软件, 浏览网站, 打开 Excel 修改表格, 统统不在话下。\\n\\n这不就是拥有最强大脑的 RPA 吗? \\n\\n突然, 我的脑海里蹦出了不少十几种优化"
5 }
6 }
7
```

分享

200 16.91 s 312 B

点击 **生成接口请求代码**, 记录刚才的请求代码和返回结果, 用于在 Cursor 中生成代码使用。

生成代码

Shell(5)

curl --location --request POST 'https://api.coze.cn/v1/workflow/run' \
--header 'Authorization: Bearer pat\_iqZN9bbC [REDACTED] Mm2JFGElpRDisM' \
--header 'User-Agent: Apifox/1.0.0 (https://apifox.com)' \
--header 'Content-Type: application/json' \
--data-raw '{
 "workflow\_id": "74 [REDACTED] 921",
 "parameters": {
 "BOT\_USER\_INPUT": "就在刚刚! 我的手机忘记关飞行模式, 深夜被微信群聊里的各种“滴滴滴”炸醒。\\n\\n原来 Anthropic 有重磅发布~\\n\\nClaude3.5 Sonnet、Claude3.5 Haiku 迎来升级。\\n\\n此外, 还引入了一项突破性的新功能—“Computer Use” 计算机使用。\\n\\n简单来说, 这项功能可以替代人工操作电脑上的网页和软件, 帮你自动化完成各种类型的工作: \\n\\n自动执行任务, 测试软件, 浏览网站, 打开 Excel 修改表格, 统统不在话下。\\n\\n这不就是拥有最强大脑的 RPA 吗? \\n\\n突然, 我的脑海里蹦出了不少十几种优化"
 }
}'

保存

保存为接口

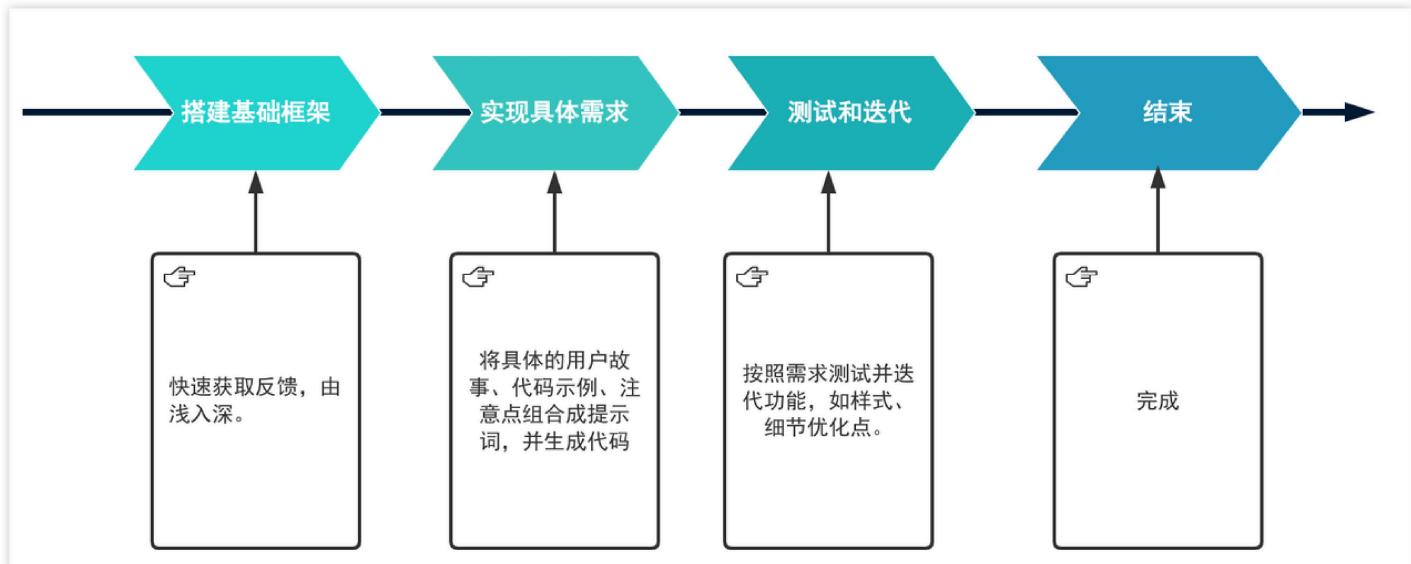
200 16.91 s 312 B

记录提示词 (大家根据实际情况记录) :

#### 四、如何在 Cursor 编程时集成 Coze 插件？

为了更快拿到反馈，我们可以先用 Cursor 生成一个基础框架，之后再逐步填充具体需求，最后进行插件的测试和迭代。

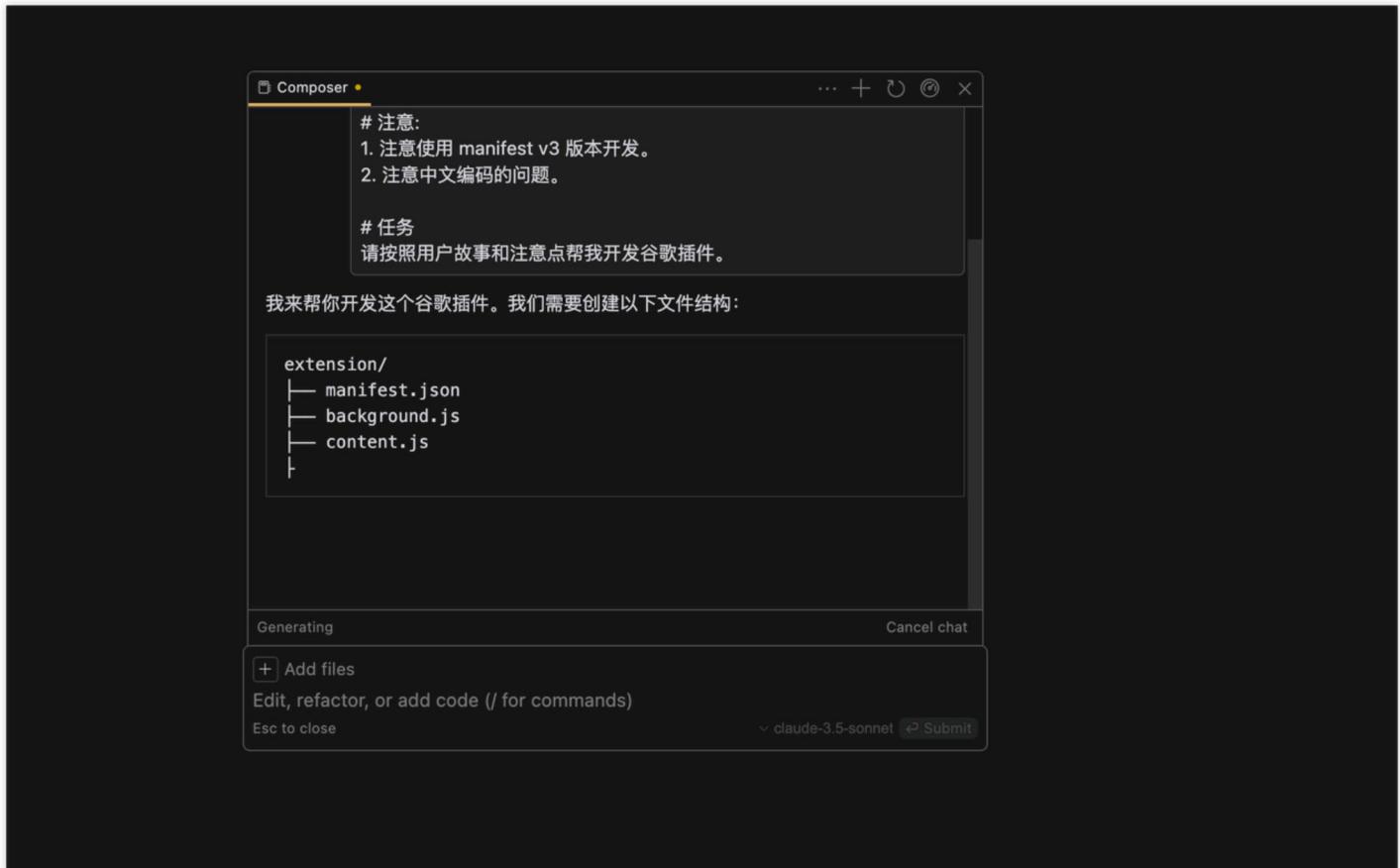
这样一来，反馈可以尽早收集，优化也会更高效。



## 一) 搭建基础框架

输入初版提示词，用于生成插件代码的基本结构。

```
1 ## 用户故事:
2 用户选中文字，可以显示浮动按钮“生成思维导图”，点击后将弹出右侧窗口，显示对应的文字。
3
4 # 注意:
5 1. 注意使用 manifest v3 版本开发。
6 2. 注意中文编码的问题。
7
8 # 任务
9 请按照用户故事和注意点帮我开发谷歌插件。
```



继续要求生成侧边栏相关的代码。

```
{ } manifest.json > ...
1 {
2 "manifest_version": 3,
3 "name": "思维导图生成器",
4 "version": "1.0",
5 "description": "将选中的文字或网页内容生成思维导图",
6 "permissions": [
7 "activeTab",
8 "scripting"
9],
10 "action": {
11 "default_popup": "popup.html"
12 },
13 "content_scripts": [
14 {
15 "matches": [""],
16 "js": ["content.js"],
17 "css": ["style.css"]
18 }
19],
20 "background": {
21 "service_workers": []
22 }
23 }
```

注意事项:

1. 使用了 Manifest V3 规范
2. 所有文件都使用 UTF-8 编码, 确保中文显示正常
3. 使用 chrome.scripting API 替代了旧版的 chrome.tabs.executeScript

要完成侧边栏的实现, 你还需要:

1. 创建一个侧边栏的 HTML 页面
2. 实现在 background.js 中打开侧边栏的逻辑
3. 添加侧边栏的样式

需要我继续完成这些功能吗?

继续

Generating

Cancel chat

+ Add files

Edit, refactor, or add code (/ for commands)

Esc to close

claude-3.5-sonnet Submit

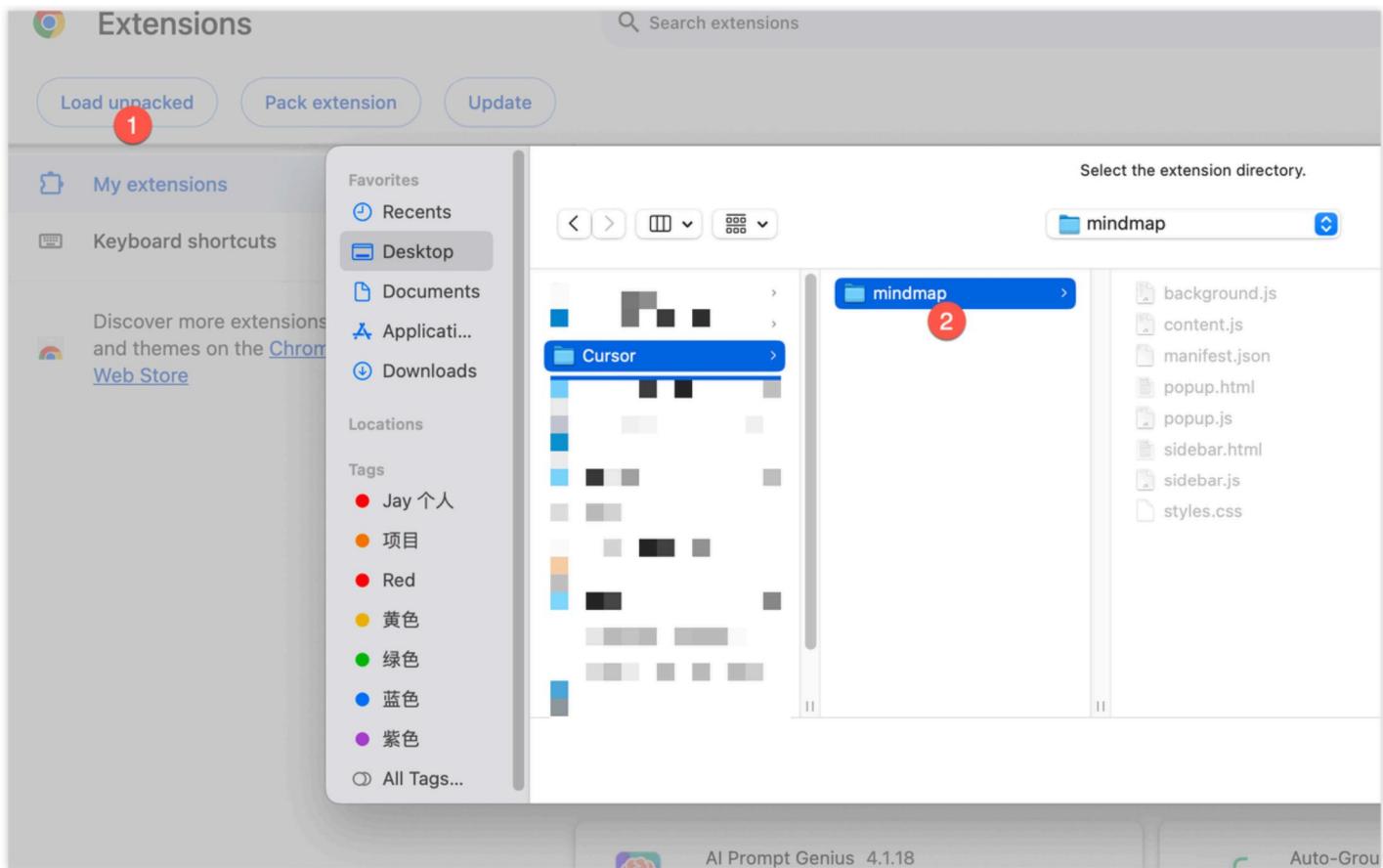
```
1 {
2 "manifest_version": 3,
3 "name": "思维导图生成器",
4 "version": "1.0",
5 "description": "将选中的文字或网页内容生成思维导图",
6 "permissions": [
7 "activeTab",
8 "scripting",
9 "tabs"
10],
11 "action": {
12 "default_popup":
13 },
14 "content_scripts":
15 "matches": ["
16 "js": ["content.js"],
17 "css": ["style.css"]
18],
19 "background": {
20 "service_worker": "
21 },
22 "web_accessible_resources": [
23 "resources": "
24 "matches": ["
25]
26 }

JS background.js 2/2 ✓
这些更新完成了以下功能：
1. 创建了一个固定在页面右侧的侧边栏
2. 侧边栏通过 iframe 实现，确保样式隔离
3. 当用户选中文本或点击生成按钮时，会在侧边栏中显示相应文本
4. 使用 postMessage 在主页面和侧边栏之间进行通信
使用方法：
1. 在页面上选中文字，点击浮动的"生成思维导图"按钮
2. 或者点击扩展图标，然后点击"生成思维导图"按钮
3. 侧边栏会显示在页面右侧，展示相应的文本内容
你可以测试一下这个实现。如果需要调整样式或添加其他功能，请告诉我。

Completed Reapply all Save all
+ Add files
Edit, refactor, or add code (/ for commands)
Esc to close claude-3.5-sonnet Submit
```

根据 AI 的提示，来进行测试。可以打开浏览器，进入插件的加载链接，选择插件代码所在的文件夹。这样一来，就能加载插件，开始后续测试啦！

插件加载链接：chrome://extensions/



在正式测试前，可以按下 **F12** 打开开发者工具，切换到 **Console** 页签。

接着点击右上角的“清除”按钮，保持 Console 页面干净，这样可以避免网页本身错误信息干扰视线。

这样在测试时，如果出现报错，Console 里显示的基本就是和插件本身相关的错误信息，方便排查问题。

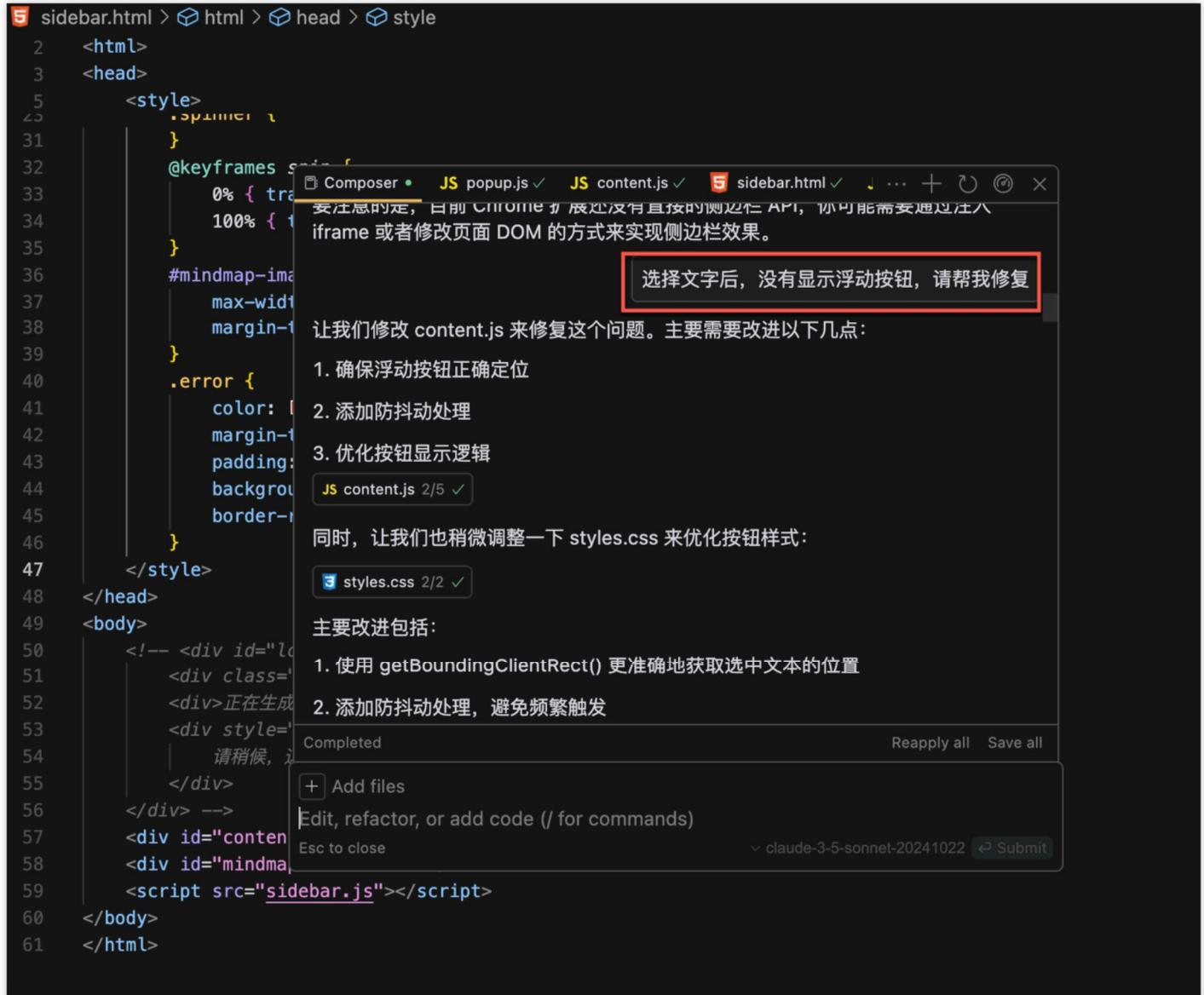
The screenshot shows the developer tools console tab. A red arrow points from the 'Filter' button to the 'Issues' section, which displays 14 issues. A red circle labeled '1' is on the 'Console' tab, and another red circle labeled '2' is on the 'Issues' button. The main content area shows a Baidu search result for '姚明' (Yao Ming) with a photo of him speaking at a conference. Below the search result, there is a summary of Yao Ming's role in the Chinese Basketball Association and a quote from his successor, Cao Zhenming.

2017年2月23日，在中国篮球协会第九届全国代表大会上，  
协主席。图为姚明在大会上致辞。新华社记者孟永民摄

接替姚明担任中国篮协主席的是国家体育总局篮球运动管理  
书记、中国篮协副主席郭振明，他自2022年初进入中国篮协，与  
郭振明表示，他对姚明最深刻的印象是敢于担当。2017年初  
是国家体育总局推进的一项重大改革举措。在没有成熟经验可资  
用自己各方面的能力、资源以及理解认识带领大家为协会建立运  
的担当。中国篮协在实体化改革之后的近8年时间里，在篮球的

在第一次测试中发现：选中文字后并未出现浮动按钮，而且 Console 中也没有任何错误提示。

这些信息可以整理后，反馈给 Cursor，以便它协助排查原因。



sidebar.html > html > head > style

```
2 <html>
3 <head>
4 <style>
5 .spanner {
6 ...
7 }
8 @keyframes ...
9 0% { tra ...
10 100% { t ...
11 }
12 #mindmap-im ...
13 max-wid ...
14 margin-t ...
15 }
16 .error {
17 color: ...
18 margin-t ...
19 padding: ...
20 backgrou ...
21 border-i ...
22 }
23 </style>
24 </head>
25 <body>
26 <!-- <div id="l ...
27 <div class=" ...
28 <div>正在生成 ...
29 <div style=" ...
30 | 请稍候，这 ...
31 </div>
32 </div> -->
33 <div id="conten ...
34 <div id="mindma ...
35 <script src="side ...
36 </body>
37 </html>
```

选择文字后，没有显示浮动按钮，请帮我修复

让我们修改 content.js 来修复这个问题。主要需要改进以下几点：

1. 确保浮动按钮正确定位
2. 添加防抖动处理
3. 优化按钮显示逻辑

JS content.js 2/5 ✓

同时，让我们也稍微调整一下 styles.css 来优化按钮样式：

styles.css 2/2 ✓

主要改进包括：

1. 使用 getBoundingClientRect() 更准确地获取选中文本的位置
2. 添加防抖动处理，避免频繁触发

Completed Reapply all Save all

+ Add files Edit, refactor, or add code (/ for commands) Esc to close claude-3-5-sonnet-20241022 Submit

第二次测试发现：选中文字后浮动按钮正常显示了，但点击按钮后侧边栏并未弹出，同时 Console 依旧没有任何错误提示。

```
{
 "manifest_version": 3,
 "name": "图片显示侧边栏",
 "version": "1.0",
 "description": "将选中文字或网页内容生成思维导图",
 "permissions": [
 "activeTab",
 "scripting",
 "tabs"
],
 "web_accessible_resources": [
 {
 "resources": ["sidebar.html"],
 "matches": ["<all_urls>"]
 }
],
 "action": {
 "default_popup": "popup.html"
 },
 "content_scripts": [
 {
 "matches": ["<all_urls>"],
 "js": ["content.js"],
 "css": ["styles.css"]
 }
],
 "background": {
 "service_worker": "background.js"
 }
}
```

按钮显示成功了，但是没有弹出侧边栏，也没有显示任何错误信息在 console

我们需要实现侧边栏功能。我们将修改几个文件来实现这个功能：

- 首先修改 manifest.json 添加必要的权限：
- 创建新文件 sidebar.html：
- 创建 sidebar.js：
- 修改 background.js：
- 修改 content.js，添加侧边栏创建和管理的代码：

Completed Reapply all Save all

+ Add files Edit, refactor, or add code (/ for commands) Esc to close claude-3-5-sonnet-20241022 Submit

反馈给 Cursor 后问题顺利修复咯！

百度一下
生成思维导图

郭振明坦言，从姚明手中接过中国篮球协会的这副担子，压力还是非常大的。“一方面，姚明主席在整个篮球界的个人影响力、整合资源的能力都是很独特的。另一方面，比照国家对三大球振兴的整体部署和要求，以及2035年建成体育强国三大球面临的任务，我们还是有一定差距的。我们的篮球核心人口有7600万，有很好的基础，而中国篮球曾经取得过的优异成绩也让大家对我们有期许，在这方面有比较大的压力和挑战。”

对于未来的工作，郭振明表示篮协班子很有信心。“我们国家有独特的体制优势，三大球振兴的战略目标也会带来政策驱动和资源整合的利好，广泛的群众基础和人才基础将是做好下一步工作的支撑。一方面，要让喜欢篮球的孩子们享受到篮球的乐趣；另一方面，也要让有天赋、有特长的孩子茁壮成长，未来能够为国争光。”

郭振明说：“我个人的能力有限，但我们中国篮协领导班子、协会全体人员、中国篮球协会的会员单位要拧成一股绳，凝心聚力，团结和带领喜欢中国篮球、热爱中国篮球的人们为中国篮球贡献力量和智慧。我们要在姚明此前工作的基础上，集思广益，营造一个充满活力的环境，继续深化篮球改革，进一步把我们的制度优势体现出来。我觉得面临的困难挑战也是我们下一步做好工作的试金石，我们有信心，也要一步一步来。”

来源：新华社

[举报/反馈](#)

郭振明坦言，从姚明手中接过中国篮球协会的这副担子，压力还是非常大的。“一方面，姚明主席在整个篮球界的个人影响力、整合资源的能力都是很独特的。另一方面，比照国家对三大球振兴的整体部署和要求，以及2035年建成体育强国三大球面临的任务，我们还是有一定差距的。我们的篮球核心人口有7600万，有很好的基础，而中国篮球曾经取得过的优异成绩也让大家对我们有期许，在这方面有比较大的压力和挑战。”

**相关推荐**



中航产融举办青年榜样培养工程 (第二季) 研讨会

看航空



中国股市最赚钱的一股的最高境界是什么

古道天涯 3评论



2024年全国羽毛球赛, 男团浙江一队

潮新闻客户端 1评论



净利润7.5%近三年海尔智家盈利能力持续提升

新浪财经 1评论



波士顿动力机器狗完成自主行走, 全程自主无遥控

量子位 1评论

在测试“选中文字，点击浮动按钮，显示侧边栏和文字”这个基础功能时，我发现 Cursor 的代码成功率不太高，经常需要花时间去调试。

为此，我准备了一个代码模板，后续遇到类似需求时，大家可以直接基于模板开发，提高效率！

手册内需要复制的内容：[AI极速赋能编程 | 03破局行动（可复制版）](#) | [直达入口](#)



## 二) 实现具体需求

在基于现有代码进行二次开发时，建议提前把已有的功能告诉 Cursor，这样可以帮助它更好地理解代码背景，从而提高开发的准确性和效率。

```
styles.css > ⚒ floating-mindmap-btn
.floating-mindmap-btn {
 width: 32px;
 height: 32px;
 background: transparent url('chrome-extension://__MSG_@@extension_id__/images/icon.svg') center/20px no-repeat;
 border: 2px solid #1e88e5;
 border-radius: 50%;
 cursor: pointer;
 transition: all 0.2s ease;
 user-select: none;
 display: flex;
 align-items: center;
 justify-content: center;
 padding: 0;
 opacity: 0.8;
 box-sizing: border-box;
 background-color: #rgba(255, 2
}

.floating-mindmap-btn:hover {
 opacity: 1;
 transform: scale(1.1);
 border-color: #1976d2;
 box-shadow: 0 2px 6px #rgba(30
}
```

现在的代码已实现功能：用户选中文字，可以显示浮动按钮“生成思维导图”，点击后将弹出右侧窗口，显示对应的文字。

很好！这是一个很好的基础功能。让我们来讨论一下可以添加的一些增强功能：

1. 复制按钮功能
  - 在侧边栏中添加一个复制按钮，方便用户复制选中的文本
  - 添加复制成功的反馈提示
2. 历史记录功能
  - 保存用户选中的文本历史记录
  - 可以在侧边栏中查看历史记录

这里是一个简单的实现示例：

**JS content.js 1/6**

**styles.css 1/9** → Tab to open

这些改进将为你的扩展添加以下功能：

Completed Reapply all Save all

+ Add files

Edit, refactor, or add code (/ for commands)  
Esc to close

claudie-3-5-sonnet-20241022 Submit

再输入完整的提示词（不要复制，注意根据个人实际代码来组合提示词）：

```
1 # 目的
2 用户想要快速基于网页内容生成一个思维导图，但是手动整理，或者手动复制到 AI 中再处理都太麻烦了。
3
4 # 用户故事
5 用户，在网页上选择一段文字，点击浮动按钮，可以生成思维导图的图片在侧边栏显示。
6
7 # 生成思维导图接口
8
9 ## 请求代码
10 ``
11 curl --location --request POST 'https://api.coze.cn/v1/workflow/run' \
12 --header 'Authorization: 自己的token' \
13 --header 'User-Agent: Apifox/1.0.0 (https://apifox.com)' \
14 --header 'Content-Type: application/json' \
15 --data-raw '{
16 "workflow_id": "自己的流程 ID",
17 "parameters": {
```

```

18 "BOT_USER_INPUT":"就在刚刚！我的手机忘记关飞行模式，深夜被微信群聊里的各种“滴滴”
19 }
20 }
21 ``
22
23 ## 请求参数介绍
24 1. parameters下的BOT_USER_INPUT: 网页内容
25 2. workflow_id为对应用户
26
27 ## 返回数据
28 ``
29 {
30 "code": 0,
31 "cost": "0",
32 "data": "{\"output\":\"https://static.shutu.cn/shutu/static/open26/2024/10/31/6a1bc6ce204e2d99e444a89c07507ef.jpeg\"}",
33 "debug_url": "https://www.coze.cn/work_flow?execute_id=743173xxxx2xx948&space_id=7431738408941223948&space_id=7393328304465412131&workflow_id=7431530629383290921",
34 "msg": "Success",
35 "token": 0
36 }
37 ``
38
39 ## 返回参数介绍
40 1. data下方的参数output为思维导图的图片地址。
41 2. code: 状态码, 不是 0 则代表出错。
42 3. msg: 返回信息
43
44 # 注意:
45 1. 注意使用 manifest v3 版本开发。
46 2. 注意中文编码的问题。
47
48 # 任务
49 根据<用户故事>和提供的<生成思维导图接口>,以及相关注意点,请优化当前谷歌插件。

```

```

1 styles.css > .floating-mindmap-btn
2
3 .floating-mindmap-btn {
4 width: 32px;
5 height: 32px;
6 background: transparent url('chrome-extension://__MSG_@@extension_id__/images/icon.svg') center/20px no-repeat;
7 border: 2px solid #1e88e5;
8 border-radius: 50%;
9 cursor: pointer;
10 transition: all 0.2s ease;
11 user-select: none;
12 display: flex;
13 align-items: center;
14 justify-content: center;
15 padding: 0;
16 opacity: 0.8;
17 box-sizing: border-box;
18 background-color: #255, 2
19 }
20
21 .floating-mindmap-btn:hover {
22 opacity: 1;
23 transform: scale(1.1);
24 border-color: #1976d2;
25 box-shadow: 0 2px 6px #255, 30
26 }

```

The screenshot shows a browser's developer tools interface. On the left, the styles.css file is open, displaying the CSS rules for the floating-mindmap-btn class. On the right, the browser's main window shows a JSON response from a POST request. The response contains a "data" key with a URL to a mindmap image, a "debug\_url", a "msg" value of "Success", and a "token". Below the JSON, there are sections for "## 返回参数介绍", "## 注意:", and "# 任务", each containing specific instructions and notes.

加载插件后，我们来测试下。

7月27日，2024年中国男子篮球职业联赛（CBA）选秀大会在北京首钢园举行。图为郭振明在选秀大会上致辞。新华社发

郭振明坦言，从姚明手中接过中国篮球协会的这副担子，压力还是非常大的。“一方面，姚明主席在整个篮球界的个人影响力、整合资源的能力都是很独特的。另一方面，比照国家对三大球振兴的整体部署和要求，以及2035年建成体育强国三大球面临的任务，我们还是有一定差距的。我们的篮球核心人口有7600万，有很好的基础，而中国篮球曾经取得过的优异成绩也让大家对我们有期待，在这方面有比较大的压力和挑战。”

对于未来的工作，郭振明表示篮协班子很有信心。“我们国家有独特的体制优势，三大球振兴的战略目标也会带来政策驱动和资源整合的利好，广泛的群众基础和人才基础将是做好下一步工作的支撑。一方面，要让喜欢篮球的孩子们享受到篮球的乐趣；另一方面，也要让有天赋、有特长的孩子茁壮成长，未来能够为国争光。”

郭振明说：“我个人的能力有限，但我们中国篮协领导班子、协会全体人员、中国篮球协会的会员单位要拧成一股绳、凝心聚力，团结和带领喜欢中国篮球、热爱中国篮球的人们为中国篮球贡献力量和智慧。我们要在姚明此前工作的基础上，集思广益，营造一个充满活力的环境，继续深化篮球改革，进一步把我们的制度优势体现出来。我觉得面临的困难挑战也是我们下一步做好工作的试金石，我们有信心，也要一步一步来。”

来源：新华社

举报/反馈

不过还是有些问题需要注意，比如：没有关闭按钮、文字不该出现在侧边栏、没有加载状态提示等。

```
{
 "manifest_version": 3,
 "name": "网页思维导图生成器",
 "version": "1.0",
 "permissions": [
 "activeTab",
 "scripting"
],
 "host_permissions": [
 "https://api.coze.net/*"
],
 "background": {
 "service_worker": "service-worker.js"
 },
 "content_scripts": [
 {
 "matches": ["<all>"],
 "js": ["content.js"],
 "css": ["styles.css"]
 }
],
 "web_accessible_resources": [
 "resources": [
 "sidebar.html"
],
 "matches": ["<all>"]
]
}
```

1. 将内联脚本移到了单独的 JS 文件中  
2. 在 manifest.json 中声明了这个新的 JS 文件  
3. 通过 `<script src="sidebar.js"></script>` 引用外部脚本文件  
这样修改后，扩展应该能够正常工作，不会再出现 CSP 相关的错误。

请优化侧边栏：1、不要显示选中的文字。2、用户点击“x”可以关闭侧边栏。3、用户选中文字或者点击“生成全文思维导图”进行处理时，侧边栏显示加载状态。

好的，我来帮你优化侧边栏的功能。首先修改 sidebar.html 和 sidebar.js，然后更新相关的样式：

sidebar.html 3/3  
JS sidebar.js Applying

然后需要修改 content

Generating.

+ Add files  
Edit, refactor, or add code (/ for commands)  
Esc to close

Cancel chat

claudie-3-5-sonnet-20241022 Submit

输入提示词，优化下交互方式，优化后的效果如下：

7月27日，2024年中国男子篮球职业联赛（CBA）选秀大会在北京首钢园举行。图为郭振明在选秀大会上致辞。新华社发

郭振明坦言，从姚明手中接过中国篮球协会的这副担子，压力还是非常大的。“一方面，姚明主席在整个篮球界的个人影响力、整合资源的能力都是很独特的。另一方面，比照国家对三大球振兴的整体部署和要求，以及2035年建成体育强国三大球面临的任务，我们还是有一定差距的。我们的篮球核心人口有7600万，有很好的基础，而中国篮球曾经取得过的优异成绩也让大家对我们有期待，在这方面有比较大的压力和挑战。”

对于未来的工作，郭振明表示篮协班子很有信心。“我们国家有独特的体制优势，三大球振兴的战略目标也会带来政策驱动和资源整合的利好，广泛的群众基础和人才基础将是做好下一步工作的支撑。一方面，要让喜欢篮球的孩子们享受到篮球的乐趣；另一方面，也要让有天赋、有特长的孩子茁壮成长，未来能够为国争光。”

郭振明说：“我个人的能力有限，但我们中国篮协领导班子、协会全体人员、中国篮球协会的会员单位要拧成一股绳、凝心聚力，团结和带领喜欢中国篮球、热爱中国篮球的人们为中国篮球贡献力量和智慧。我们要在姚明此前工作的基础上，集思广益，营造一个充满活力的环境，继续深化篮球改革，进一步把我们的制度优势体现出来。我觉得面临的困难挑战也是我们下一步做好工作的试金石，我们有信心，也要一步一步来。”

来源：新华社

举报/反馈

目前还有很多可以优化的地方，比如缺少思维导图的查看和下载功能，样式设计上也不够美观。

我们可以让 Cursor 继续进行优化，这里就不再具体演示了。

最终效果如下：

为郭振明在选秀大会上致辞。新华社发

郭振明坦言，从姚明手中接过中国篮球协会的这副担子，压力还是非常大的。“一方面，姚明主席在整个篮球界的个人影响力、整合资源的能力都是很独特的。另一方面，比照国家对三大球振兴的整体部署和要求，以及2035年建成体育强国三大球面临的任务，我们还是有一定差距的。我们的篮球核心人口有7600万，有很好的基础，而中国篮球曾经取得过的优异成绩也让大家对我们有期待，在这方面有比较大的压力和挑战。”

对于未来的工作，郭振明表示篮协班子很有信心。“我们国家有独特的体制优势，三大球振兴的战略目标也会带来政策驱动和资源整合的利好，广泛的群众基础和人才基础将是做好下一步工作的支撑。一方面，要让喜欢篮球的孩子们享受到篮球的乐趣；另一方面，也要让有天赋、有特长的孩子茁壮成长，未来能够为国争光。”

郭振明说：“我个人的能力有限，但我们中国篮协领导班子、协会全体人员、中国篮球协会的会员单位要拧成一股绳、凝心聚力，团结和带领喜欢中国篮球、热爱中国篮球的人们为中国篮球贡献力量和智慧。我们要在姚明此前工作的基础上，集思广益，营造一个充满活力的环境，继续深化篮球改革，进一步把我们的制度优势体现出来。我觉得面临的困难挑战也是我们下一步做好工作的试金石，我们有信心，也要一步一步来。”

来源：新华社

举报/反馈

**相关推荐**

- 专题学习党的二十届三中全会精神——区司法局召开党委...
- 澎湃新闻客户端 1评论
- 《“纯血”鸿蒙：华为的挑战与机遇》
- 墨而不染 1评论
- 韩国5100万人口，汽车保有量2500万辆，山东1亿人保...
- 文史群英会 1评论
- 姚明辞职了，功过自有后人评说
- 极目新闻 1评论

完整代码：

手册内需要复制的内容：[AI极速赋能编程 | 03破局行动（可复制版）](#) | [直达入口](#)



## 五、总结

如今的 AI 时代，与其单打独斗，不如学会借力。

Coze 插件就像是一个个现成的乐高积木，而我们要做的，就是用创意把它们拼接成独特的作品。

不必执着于从零开始写代码，学会借助现有的工具，让创意为你带来无限可能。

毕竟，在这个快速迭代的 AI 时代，懂得站在巨人肩膀上的人，总能看得更远，走得更快。

### 6.3 3分钟部署上线小产品？Cursor 和 Devbox 让开发从未如此简单！

身边越来越多人尝试用 Cursor 写代码、开发小产品了。

如果想要实现商业化或引流，我们的小工具和小产品必须发布到互联网上，这样才能被更多人看到并使用。

可一到要部署上线，很多非技术的朋友们都抓瞎了：

不仅要安装 Python，还要设置各种复杂的编译环境参数，  
然后是云服务器要采购、配置、部署代码，  
最后还要搞什么 https 证书...

有人可能会告诉你：“别担心，百度、AI 搜索，能找到一堆教程的”。

说实话，先不说他们愿不愿意去搜，即使搜到了也很难明白。

今天我要告诉你一个好消息：不用记命令、不用配环境、不用碰服务器，3分钟就能轻松把项目部署上线。

想知道是什么神仙操作吗？接着往下看教程 ↗

## 一、安装

在安装前，我们需要做一些准备事项：

- 科学上网
- 本地安装 Cursor，可参考文章：

第一步，打开网址，登录和注册：<https://cloud.sealos.run/?uid=ry2TN0LDRb>

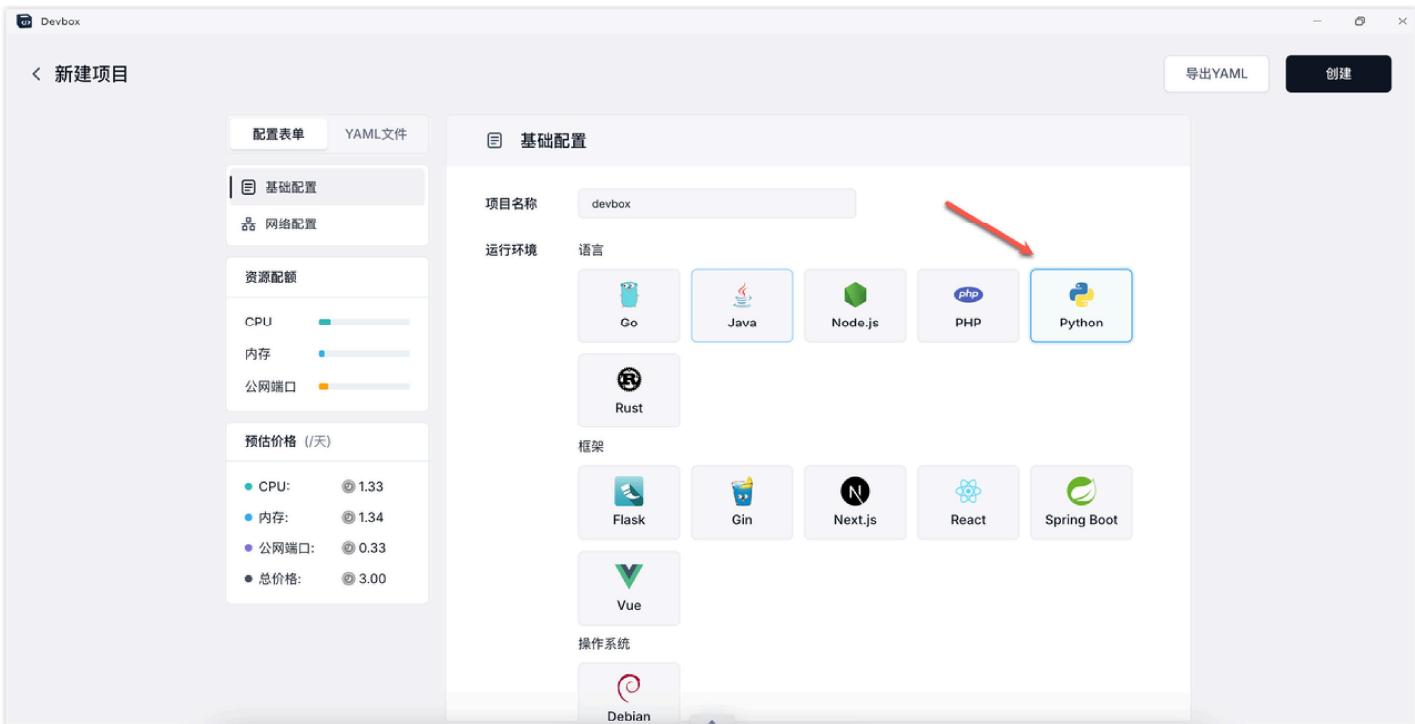


登录完成后，进入首页，点击 **Devbox**。

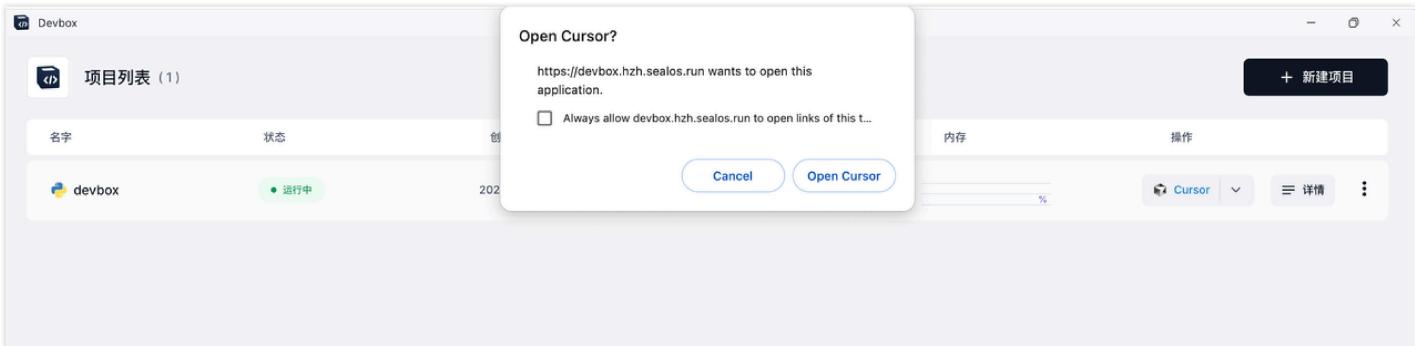
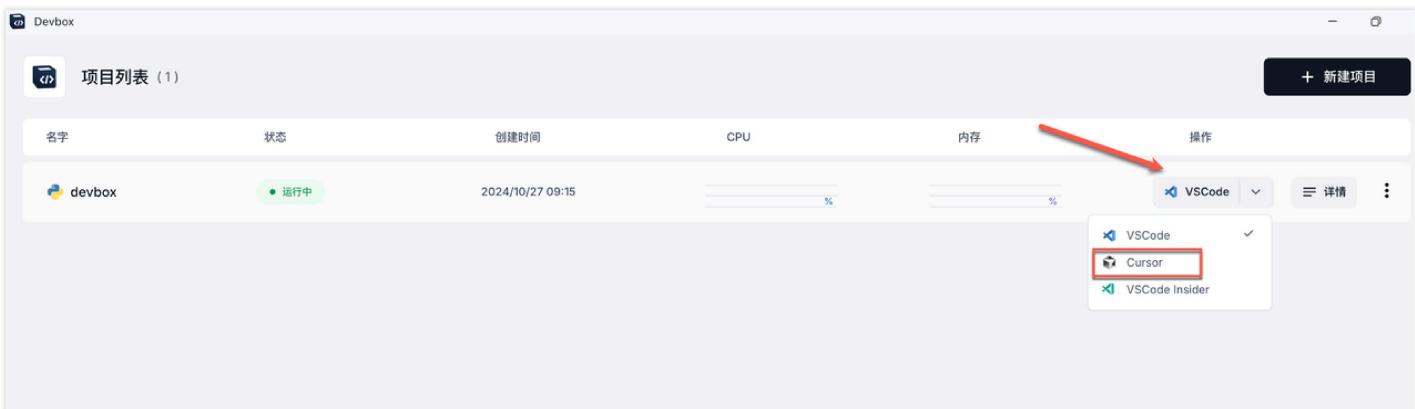
The image shows the Sealos Cloud homepage. On the left, there is a sidebar with monitoring metrics like CPU and memory usage, and a warning section. The main area is titled "所有应用" (All Applications) and lists various services: Application Management, Cost Center,定时任务 (Scheduled Tasks), Database, KubePanel, Object Storage, Cloud Development, Document Center, Application Store, Terminal, Work Orders, FastGPT, Invitation Registration, and Private Cloud. A red arrow points to the "Devbox" icon, which is highlighted with a red border. On the right side, there is a user profile section with options like Logout, Account Settings, Tickets, Personal Space, and kubeconfig. Below that is a financial summary showing a balance of 5.00 and usage statistics for the previous and next month.

过去 🐾 熊猫 Jay一直推荐大家学习 Python，这里以 Python 项目为例来演示。

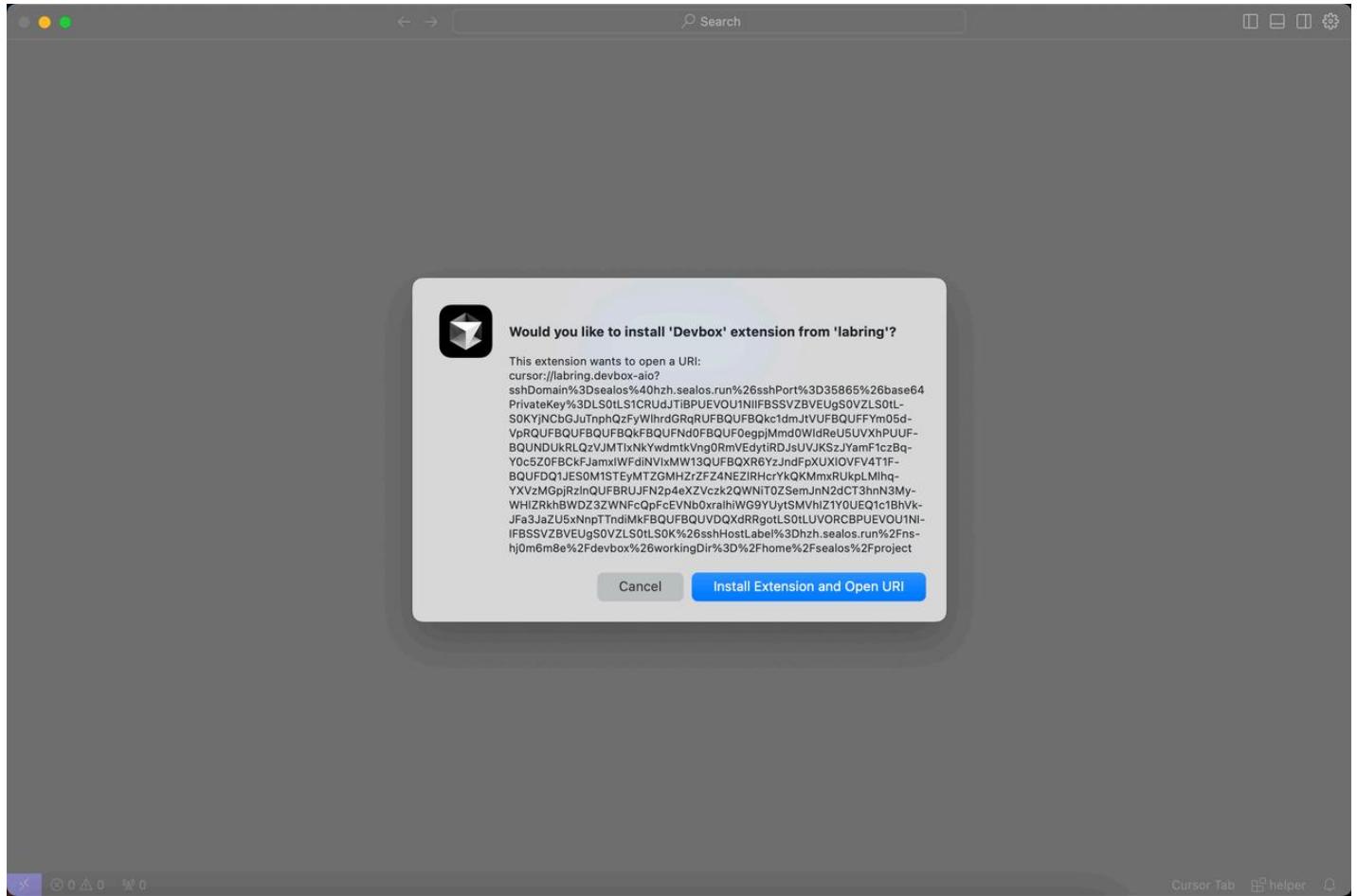
除此之外，还支持 **Go、Java、Node.js、PHP、Rust** 等主流的编程语言和框架。



创建项目后，在操作列选择 Cursor 后，记得要单击它。



点击 **Open Cursor** 后，将会自动打开本地的 Cursor 编程工具。再点击 **Install Extension and Open URI**，安装必备的一些插件。



我们需要等待推荐的插件都安装完成，这里可能需要花费一些时间。插件出现 **Disable** 或者 **Uninstall** 的按钮则代表安装成功。

The screenshot shows the Visual Studio Code extension marketplace. The main focus is the "Python" extension by ms-python, version v2024.12.3. The extension has over 140 million installs and a 4.5-star rating. It provides Python language support with IntelliSense, Pylance, debugging, and more. The sidebar on the left shows other recommended extensions like i18n Ally, Code Runner, and Excalidraw. The bottom status bar indicates an SSH connection to a devbox.

**Python extension for Visual Studio Code**

A Visual Studio Code extension with rich support for the Python language (for all actively supported Python versions), providing access points for extensions to seamlessly integrate and offer support for IntelliSense (Pylance), debugging (Python Debugger), formatting, linting, code navigation, refactoring, variable explorer, test explorer, and more!

**Support for vscode.dev**

The Python extension does offer some support when running on vscode.dev (which includes github.dev). This includes partial IntelliSense for open files in the editor.

**Installed extensions**

The Python extension will automatically install the following extensions by default to provide the best Python development experience in VS Code:

- Pylance - to provide performant Python language support

**Categories**

- Programming Languages
- Debuggers
- Other
- Data Science
- Machine Learning

**Resources**

- Marketplace
- Issues
- Repository
- License
- ms-python

**More Info**

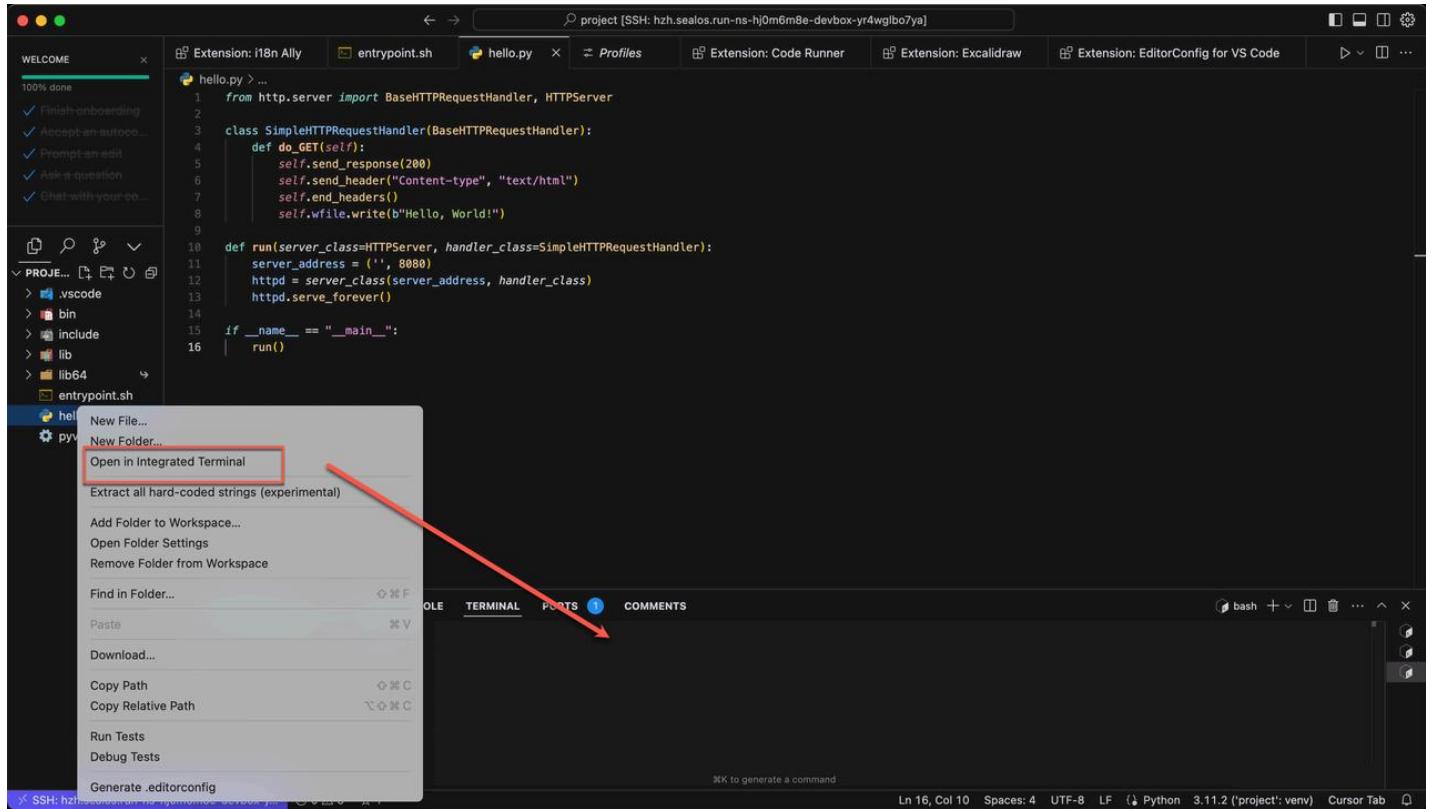
Published 2016-01-19, 23:03:11  
Last updated 2024-10-23

## 二、初体验

在绑定了 Devbox 项目的 Cursor 界面，我们看到了初始化的代码结构。

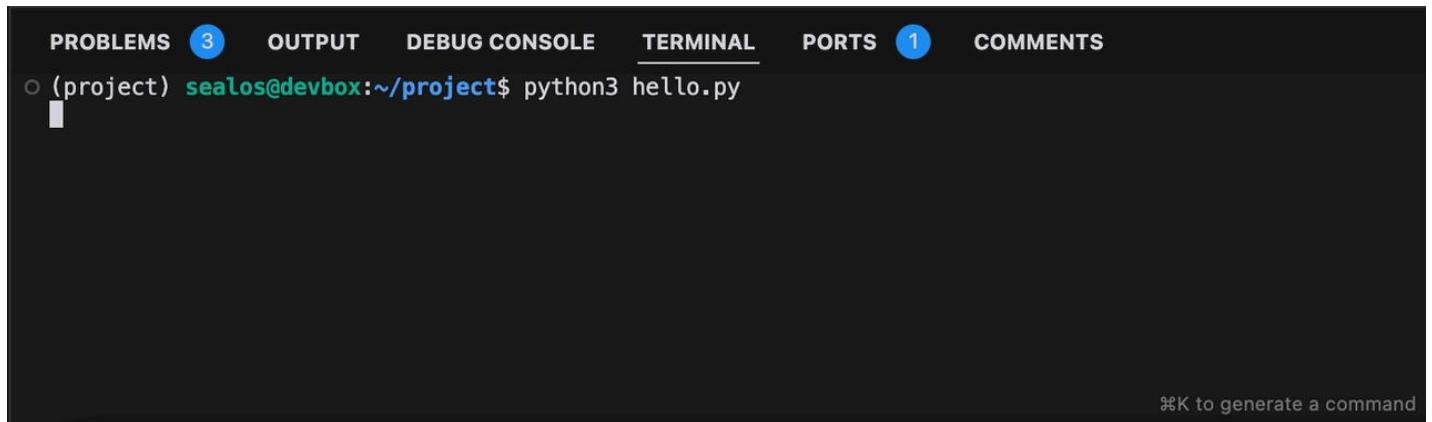
我们来演示下如何启动代码。

我们在左下方右击空白处，点击 Open in integrated Terminal 打开控制台。

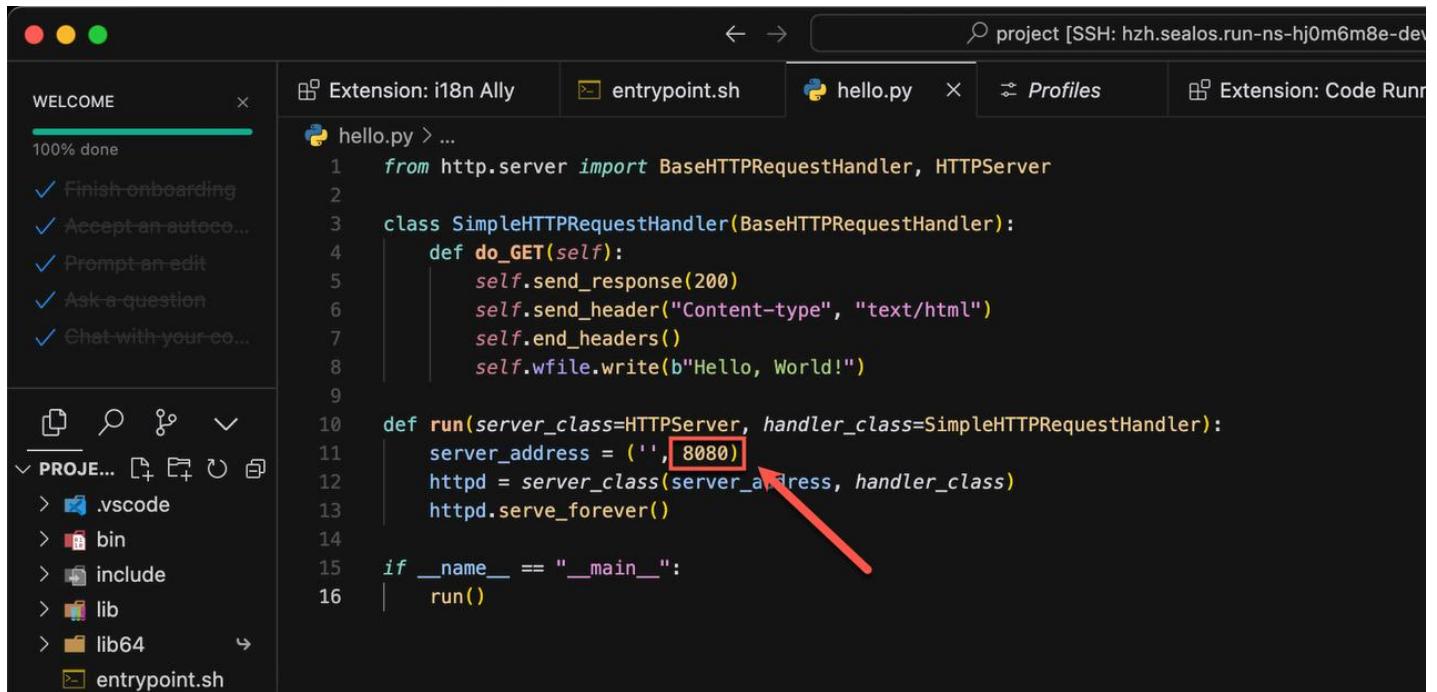


输入命令：**python3 hello.py**，不出现任何报错，则代表启动成功。

其他编程语言，或者框架，具体启动命令以实际为准，如果不清楚，可以直接问 Cursor。



我们打开本地浏览器，查看下运行效果。这里的端口是 8080，那么地址为：<http://localhost:8080>



```

WELCOME ×
Extension: i18n Ally | entrypoint.sh | hello.py | Profiles | Extension: Code Runn...
100% done
✓ Finish onboarding
✓ Accept an autoco...
✓ Prompt an edit
✓ Ask a question
✓ Chat with your co...
PROJ... .vscode bin include lib lib64 entrypoint.sh

hello.py > ...
1 from http.server import BaseHTTPRequestHandler, HTTPServer
2
3 class SimpleHTTPRequestHandler(BaseHTTPRequestHandler):
4 def do_GET(self):
5 self.send_response(200)
6 self.send_header("Content-type", "text/html")
7 self.end_headers()
8 self.wfile.write(b"Hello, World!")
9
10 def run(server_class=HTTPServer, handler_class=SimpleHTTPRequestHandler):
11 server_address = ('', 8080)
12 httpd = server_class(server_address, handler_class)
13 httpd.serve_forever()
14
15 if __name__ == "__main__":
16 run()

```



这次的分享主题是「代码快速上线」，那这个 demo 在互联网上怎么查看效果呢？

我们回到 Devbox 项目中，点击外网地址，就可以在公网看到效果啦。

The screenshot shows the Sealos DevBox management interface for a box named 'devbox'. The left sidebar includes sections for basic information, SSH configuration, and events. The main area has tabs for real-time monitoring (CPU 0.08%, Memory 37.63%) and network configuration (1 interface). A red box highlights the '外网地址' (External IP) field, which shows 'https://pcjwiebxzwfn.hzh.sealos.run'. Below this is a browser preview window showing the application's output.

The screenshot shows a browser window displaying the application's output. The address bar shows the external URL: 'https://pcjwiebxzwfn.hzh.sealos.run'. The page content displays the text 'Hello, World!'. The browser interface includes standard navigation buttons (back, forward, search).

当然，这只是用初始化的代码演示的效果。接下来，我们来展示下修改部分代码后，怎么更新线上环境。

这里以修改网页内容为例，加上 **I am Jay** 的字符，记得保存代码。

```

hello.py > ...
1 from http.server import BaseHTTPRequestHandler, HTTPServer
2
3 class SimpleHTTPRequestHandler(BaseHTTPRequestHandler):
4 def do_GET(self):
5 self.send_response(200)
6 self.send_header("Content-type", "text/html")
7 self.end_headers()
8 self.wfile.write(b"Hello, World! I am Jay") I am Jay
9
10 def run(server_class=HTTPServer, handler_class=SimpleHTTPRequestHandler):
11 server_address = ('', 8080)
12 httpd = server_class(server_address, handler_class)
13 httpd.serve_forever()
14
15 if __name__ == "__main__":
16 run()

```

在控制台 (Terminal) 按 **Ctrl + C** 关闭应用，再输入命令 **python3 hello.py** 重新启动。

```

⑧ (project) sealos@devbox:~/project$ python3 hello.py
^CTraceback (most recent call last):
 File "/home/sealos/project/hello.py", line 16, in <module>
 run()
 File "/home/sealos/project/hello.py", line 13, in run
 httpd.serve_forever()
 File "/usr/lib/python3.11/socketserver.py", line 233, in serve_forever
 ready = selector.select(poll_interval)
 ^^^^^^^^^^^^^^^^^^^^^^^^^^
 File "/usr/lib/python3.11/selectors.py", line 415, in select
 fd_event_list = self._selector.poll(timeout)
 ^^^^^^^^^^^^^^^^^^^^^^^^^^
KeyboardInterrupt

```

○ (project) sealos@devbox:~/project\$ █

无论本地还是线上都会立刻更新成功，不需要做任何多余的动作，线上代码就生效了。

但是，我们并没有完成线上发布哦，毕竟电脑不可能一直运行程序，后续步骤会介绍。



### 三、创建 AI 聊天页面

「Hello World」，毕竟太简单了。

接下来，我们以一个真实有效的小案例 -- **接入智谱大模型的AI 聊天工具** 来展示下全过程。

#### 一) 提前准备事项

因为需要接入智谱的 API，所以我们需要提前做一些准备：

- 智谱 API 接入案例代码
- 智谱 API Key
- 本地安装智谱 SDK

##### 1、智谱 API 接入案例代码：

| 地址：<https://www.bigmodel.cn/dev/api/normal-model/glm-4>

BigModel 控制台 文档

使用指南 接口文档 活动中心 最佳实践 常见问题

开发指南

> SDK 开发

HTTP 调用

HTTP 请求参数

HTTP 用户鉴权

> 第三方框架

开放接口

语言模型

GLM-4 系列

GLM-4V 系列

> 文生图模型

> 文生视频模型

> Agent模型

> 代码模型

帮助中心

技术社群

财务管理 API 密钥 个人中心

本页面录

同步调用

请求参数

message 格式

响应参数

请求示例

流式输出

响应参数

请求示例

函数调用

异步调用

请求参数

响应参数

调用示例

任务结果查询

请求参数

响应参数

请求示例

我们已经升级到最新的模型GLM-4和GLM-3-Turbo，支持系统提示、函数调用、检索、Web\_Search等新功能。要使用这些新功能，需要升级到最新版本的Python SDK。如果您安装了旧版本的SDK，请更新到最新版本。

```
1 pip install --upgrade zhipuai
```

```
1 from zhipuai import ZhipuAI
2 client = ZhipuAI(api_key='') # 请填写您自己的APIKey
3 response = client.chat.completions.create(
4 model="glm-4", # 请填写您要调用的模型名称
5 messages=[
6 {"role": "system", "content": "你是一个乐于回答各种问题的小助手。你的任务是提供专业、准确、有洞察力的建议。"},
7 {"role": "user", "content": "我对太阳系的行星非常感兴趣，尤其是土星。请提供关于土星的基本信息，包括它的大小、组成
8],
9 stream=True,
10)
11 for chunk in response:
12 print(chunk.choices[0].delta)
```

响应示例:

```
1 data: {"id": "8313807536837492492", "created": 1706092316, "model": "glm-4", "choices": [{"index": 0, "delta": {"role": "assistant", "content": "土星是太阳系中第六大行星，也是离太阳第七远的行星。它以其巨大的体积和美丽的环系而闻名。"}]}
2 data: {"id": "8313807536837492492", "created": 1706092316, "model": "glm-4", "choices": [{"index": 0, "delta": {"role": "assistant", "content": "土星的直径大约是地球的十倍，质量则是地球的约95倍。它的主要成分为氢气和氦气，内部结构类似于木星。"}]}
```

基于代码案例和需求，编写简单的提示词。

复制

## Markdown

1 帮我根据下方的代码请求和响应示例，创建一个 AI 聊天页面。

2

3 需求：用户在页面聊天窗口输入信息，点击发送后，将收到 AI 流式输出的回复效果。

4

5 代码请求示例：

```
6 from zhipuai import ZhipuAI
7 client = ZhipuAI(api_key="") # 请填写您自己的APIKey
8 response = client.chat.completions.create(
9 model="glm-4", # 请填写您要调用的模型名称
10 messages=[
11 {"role": "system", "content": "你是一个乐于回答各种问题的小助手，你的任务是提
12 {"role": "user", "content": "我对太阳系的行星非常感兴趣，尤其是土星。请提供关
13],
14 stream=True,
15)
16 for chunk in response:
17 print(chunk.choices[0].delta)
18
```

19 代码响应示例：

20 ````

```
21 data: {"id": "8313807536837492492", "created": 1706092316, "model": "glm-4", "choice
22 data: {"id": "8313807536837492492", "created": 1706092316, "model": "glm-4", "choice
23
24 data: {"id": "8313807536837492492", "created": 1706092316, "model": "glm-4", "choice
25 data: {"id": "8313807536837492492", "created": 1706092316, "model": "glm-4", "choice
26 data: {"id": "8313807536837492492", "created": 1706092316, "model": "glm-4", "choice
27 data: [DONE]
28 ````
```

## 2、生成并复制 API Key

地址: <https://bigmodel.cn/usercenter/apikeys>

The screenshot shows the 'API keys' section of the BigModel control panel. On the left sidebar, 'API keys' is selected. The main area displays a table of existing API keys with columns: 名称 (Name), API key, 创建时间 (Created Time), 上次使用时间 (Last Used Time), and 操作 (Operation). The table contains four entries. A red arrow points to the blue '+ 添加新的API Key' button in the top right corner.

| 名称            | API key  | 创建时间                | 上次使用时间     | 操作  |
|---------------|----------|---------------------|------------|-----|
| 系统默认          | 14C...k  | 2024-01-16 20:32:06 | 未使用        | ... |
| 系统默认 (体验中心调用) | ebf...vQ | 2024-03-18 18:16:30 | 2024-09-14 | ... |
|               | d11...ZM | 2024-09-04 10:25:15 | 2024-09-19 | ... |
| chrome        | bbf...A  | 2024-09-15 17:06:34 | 2024-10-27 | ... |

### 3、启动文件中加入智谱 SDK

为了确保接下来能够正常上线，我们需在进入 Devbox 绑定的 Cursor 代码里的启动文件 -- **entrypoint.sh**，添加智谱 SDK 的安装命令。

```
1 pip install --upgrade zhipuai -i https://mirrors.aliyun.com/pypi/simple/
```

**entrypoint.sh** 完整内容如下：

```
1 #!/bin/bash
2 python3 -m venv venv # 创建虚拟环境
3 source venv/bin/activate # 激活虚拟环境
4 pip install --upgrade zhipuai -i https://mirrors.aliyun.com/pypi/simple/
5 python3 hello.py
```

## 二) 基于 Cursor 生成 AI 聊天页面

打开 Composer，输入提前准备的提示词。

```
hello.py > ...
1 from http.server import BaseHTTPRequestHandler, HTTPServer
2
3 class SimpleHTTPRequestHandler(BaseHTTPRequestHandler):
4 def do_GET(self):
5 self.send_response(200)
6 self.send_header("Content-type", "text/html")
7 self.end_headers()
8 self.wfile.write(b"Hello, World! I am Jay")
9
10 def run(server_class=HTTPServer, handler_class=SimpleHTTPRequestHandler):
11 server_address = ('', 8080)
12 httpd = server_class(server_address, handler_class)
13 httpd.serve_forever()
14
15 if __name__ == "__main__":
16 run()
```



配置提前准备好的 API Key，并确定所使用的智谱模型版本。**glm-4-flash** 是免费的，其他模型官方也提供了免费额度。

```
WELCOME
100% done
✓ Finish-onboarding
✓ Accept-an-autocom...
✓ Prompt-en-edit
✓ Ask-a-question
✓ Chat-with-your-co...
PROJECT [SSH: HZH.SE...
> .vscode
> bin
> include
> lib
> lib64
> share
 chat.html
 entrypoint.sh
 hello.py
 index.html
 pyvenv.cfg

hello.py > ChatHandler > do_POST
1
client = ZhipuAI(api_key="bb564...irA") # 请填写您自己的APIKey
2
class ChatHandler(BaseHTTPRequestHandler):
 def do_GET(self):
 if self.path == '/':
 self.send_response(200)
 self.send_header("Content-type", "text/html")
 self.end_headers()
 with open('chat.html', 'rb') as file:
 self.wfile.write(file.read())
 else:
 self.send_error(404)

 def do_POST(self):
 if self.path == '/chat':
 content_length = int(self.headers['Content-Length'])
 post_data = self.rfile.read(content_length)
 user_message = json.loads(post_data.decode('utf-8'))['message']

 self.send_response(200)
 self.send_header('Content-Type', 'text/event-stream')
 self.send_header('Cache-Control', 'no-cache')
 self.end_headers()

 async def stream_response():
 response = client.chat.completions.create(
 model="glm-4",
 messages=[{"role": "system", "content": "你是一个乐于回答各种问题的小助手，你的任务是提供专业、准确、有洞察力的建议。"}, {"role": "user", "content": user_message}], stream=True,
```

控制台输入命令 `./entrypoint.sh` 启动项目，打开网页查看效果。

## AI 聊天

用户：你是什么版本的大模型

AI：我是智谱清言，一个基于语言模型的人工智能助手。我可能没有你所提到的“大模型”那么大的规模，但我在2023之前接受了大量的数据训练，能够提供专业、准确、有洞察力的建议和回答。如果你有任何问题，无论是科学、技术、生活、文化等领域，我都会尽力为你提供帮助。

输入您的问题...

发送

AI 正常回复了，但是发现一个问题，没有体现流式输出的效果。重新让 Cursor 优化下。

hello.py

很好，整体测试成功了，但是 AI 的回复在页面没有体现流式输出的效果。

我理解您的问题。确实，当前的实现虽然在服务器端使用了流式输出，但在客户端并没有实时显示 AI 的回复。让我们修改一下 `chat.html` 文件，以实现流式显示 AI 回复的效果。

以下是 `chat.html` 文件的修改建议：

chat.html 2/2 •

这些修改主要集中在 JavaScript 部分，主要变化如下：

1. 修改了 `addMessage` 函数，现在它会返回一个用于显示 AI 响应的 `span` 元素。
2. 在处理 AI 响应时，我们不再累积完整的响应，而是直接将每个接收到的数据片段添加到 AI 响应元素中。
3. 移除了 `aiResponse` 变量，因为我们现在是实时显示响应。

再重启下。这次发现 AI 回复的效果实现流式输出了，但是用户的聊天内容丢失了。

# AI 聊天

用户：

AI：我是智谱清言，一个基于语言模型的人工智能助手。我的背后是清华大学 KEG 实验室和智谱 AI 公司于 2023 年共同训练的语言模型 GLM-130B。我会根据你的问题，尽力提供专业、准确、有洞察力的建议。如果你有任何问题，都可以向我提出。

用户：

AI：鲁迅（1881年9月25日 – 1936年10月19日），原名周树人，是中国近现代著名的文学家、思想家、革命家。他在中国现代文学史上有着举足轻重的地位，被认为是现代汉语文学的奠基人之一。鲁迅的作品主要包括短篇小说集《呐喊》和《彷徨》，杂文集《坟》、《热风》、《华盖集》等。他的文学作品以其深刻的社会批判性和艺术性著称，他通过作品揭示了社会的不公和旧社会的黑暗，倡导民主和科学，反对封建主义和迷信。他的许多作品都表达了对底层人民的同情以及对旧社会制度的批判。的思想深受启蒙和批判精神的影响，他提倡使用白话文写作，反对文言文的束缚，对中国文学语言的发展产生了深远的影响。此外，鲁迅还翻译了许多外国文学作品，对中外文化交流作出了贡献。不仅仅是一位文学家，他还是一位社会活动家，参与了中国共产党的外围组织“左翼作家联盟”的活动，通过其作品和行动支持中国的社会变革。他的思想和作品至今仍然具有很高的研究价值和社会影响力。

输入您的问题...

发送

修复 Bug 时，我们尽量做到详细描述复现步骤和错误症状，输入整理后的提示词，请 Cursor 修复。

很好，页面实现了AI回复时流式输出的效果，但是用户输入的内容，点击发送后，在聊天窗口里不显示了，这样无法追溯。

我理解您的问题。确实，用户输入的内容在发送后应该保留在聊天窗口中，以便于追溯对话历史。让我们修改一下 `chat.html` 文件来解决这个问题。

以下是 `chat.html` 文件的修改建议：

主要的变化在 `addMessage` 函数和事件监听器中：

1. 修改了 `addMessage` 函数：
  - 现在它为所有消息（包括用户消息和 AI 响应）创建相同结构的 HTML 元素。
  - 消息内容被放在 `<span>` 标签内，这样可以统一处理用户消息和 AI 响应。
2. 在发送用户消息时：
  - 使用 `addMessage('用户', message)` 来显示用户的输入，确保用户消息会显示在聊天窗口中。

Completed Reapply all Save all Reject all Accept all

+ `hello.py` ×

Edit, refactor, or add code (/ for commands)

Esc to close claude-3.5-sonnet Submit

这次修复成功了，用户聊天记录保留了，且 AI 回复也实现了流式输出的效果。



最后，整体功能就绪后，再优化下页面。

The screenshot shows a code editor with three tabs: 'Composer' (yellow), 'chat.html' (orange), and 'hello.py' (green). A message at the top says 'Checkpoint 3 saved. checkout'. Below it, a preview window for 'hello.py' shows a blue Python icon and the text '页面不够美观, 请整体优化下, 使用蓝色和白色的配色。' (The page is not very aesthetically pleasing, please overall optimize it, use a blue and white color scheme.). The main editor area has a title bar 'chat.html 4/4'. The code content discusses improvements to the chat interface.

```

好的, 我理解您希望优化页面的美观度, 并使用蓝色和白色的配色方案。我们可以通过修改 CSS 样式来实现这一点。以下是优化后的 chat.html 文件内容:

这个新版本的 chat.html 文件主要做了以下改进:
1. 使用了蓝色 (#1e90ff) 和白色作为主要配色。
2. 改进了整体布局, 使用了圆角和阴影效果, 增加了视觉深度。
3. 优化了字体和间距, 提高了可读性。
4. 美化了输入框和发送按钮的样式, 增加了悬停效果。
5. 调整了聊天容器的样式, 使消息更加清晰。

```

Completed      Reapply all    Save all    Reject all    **Accept all**

+ `hello.py` ×

Edit, refactor, or add code (/ for commands)

Esc to close      claude-3.5-sonnet      Submit

搞定!



### 三) 发布上线

- 首先进行 Devbox 项目详情, 点击发布版本。

基础信息

名字: devbox

镜像源 (私有): hub.hzh.sealos.run/ns-hj0m6m9e/devbox

创建时间: 2024-10-27 09:15

启动环境: python-3.10

启动时长: 5h8m

CPU Limit: 2 Core

Memory Limit: 4 G

SSH 配置

连接串: ssh -p 35865 sealos

私钥:

事件

最近错误: 无

实时监控 (更新时间 14:24)

CPU 0.10% 内存 37.22%

网络配置 (1)

内网地址: http://de... 端口: 35865  
外网地址: https://pcj... 端口: 8080 os.run

版本历史

暂无版本 发布版本

填写本地发版的版本号和描述信息。

实时监控 (更新时间 14:25)

发布版本

镜像名: hub.hzh.sealos.r /devbox

版本配置

版本号: v1.0.0

版本描述: AI 聊天页面版本发布

发版

注意, 发布将暂时停止 Devbox, 发版后会自动启动。



稍等一会，在版本历史界面我们点击 **上线**，将会跳转到正式环境的配置界面。

| 版本号    | 状态 | 创建时间             | 版本描述        | 操作                                                             |
|--------|----|------------------|-------------|----------------------------------------------------------------|
| v1.0.0 | 成功 | 2024-10-27 14:27 | AI 聊天页面版本发布 | <span style="border: 2px solid red; padding: 2px;">上线</span> ⋮ |

2、按照需求配置正式环境的硬件参数。

基础配置

应用名称: devbox-release

镜像源: 公共

部署模式: 固定实例

CPU: 2 (Core)

内存: 4 G

网络配置

容器暴露端口: 8080

开启公网访问: 开启

如果未来，我们的网站流量过大，我们可以点击「部署模式」为 弹性伸缩，这样服务器会根据流量和负载的情况来扩充配置，避免程序宕机。

部署模式

固定实例 弹性伸缩

CPU: 50 %

实例数: 5

CPU: 2 (Core)

内存: 4 G

前面这段话和弹性伸缩相关的图片，对于非技术的朋友可能会有点难以理解。

### 举个例子：

某宝在双十一之前，流量一般，所以硬件要求不高，一台服务器就够了。

但是双十一到了，为了分担流量高峰所带来的压力，保证软件应用不挂掉，系统会自动扩充到 100台服务器。

双十一过去后，流量慢慢变小了，这时候为了避免浪费资源，系统自动缩小到 5 台甚至更少的服务器。

这样的部署方式就是「弹性伸缩」。

### 3、配置网络（可选）

官方提供的域名没什么特定含义。我们可以自定义网站域名。若不需要，可跳过此步骤。

## 网络配置

容器暴露端口：8080，开启公网访问开关已打开。  
自定义域名输入框被红色方框标记。

## 高级配置

启动命令：① 若不填写，将使用默认命令。  
运行命令：/bin/bash -c  
命令参数：/home/sealos/project/entrypoint.sh

## 环境变量

编辑环境变量按钮。



进入我们自己的阿里云域名列表，选择一个域名，点击右侧的解析。

| 操作                                                           | 域名                 | 状态 | 域名持有者 | 注册时间                | 到期时间                |
|--------------------------------------------------------------|--------------------|----|-------|---------------------|---------------------|
| <a href="#">续费</a>   <a href="#">解析</a>   <a href="#">管理</a> | aivideossora.top   | 正常 |       | 2024-02-22 09:53:22 | 2025-02-22 09:53:22 |
| <a href="#">续费</a>   <a href="#">解析</a>   <a href="#">管理</a> | aivideos-sora.com  | 正常 |       | 2024-02-22 09:53:22 | 2025-02-22 09:53:22 |
| <a href="#">续费</a>   <a href="#">解析</a>   <a href="#">管理</a> | ai-sora.shop       | 正常 |       | 2024-02-21 22:48:55 | 2025-02-22 07:59:59 |
| <a href="#">续费</a>   <a href="#">解析</a>   <a href="#">管理</a> | sora-prompt.com.cn | 正常 |       | 2024-02-21 22:48:52 | 2025-02-21 22:48:52 |
| <a href="#">续费</a>   <a href="#">解析</a>   <a href="#">管理</a> | sora-prompt.cn     | 正常 |       | 2024-02-21 22:48:52 | 2025-02-21 22:48:52 |

添加一条记录。

记录类型 选择 CNAME，主机记录 根据喜好填写一个自定义的二级域名，记录值 填写 Devbox 提供的域名，最后确定。



为了检查配置是否生效，点击新增记录右侧的 **生效检测**。当解析结果和配置的记录值一致，则代表成功了。

| 探测点    | DNS服务器        | 解析时间      | 解析结果              | 预期解析结果 | 操作                                                                     |
|--------|---------------|-----------|-------------------|--------|------------------------------------------------------------------------|
| 贵州贵阳移动 | 211.139.5.29  | 137.037ms | uexw: [■■■■■] run | -      | <a href="#">详情</a>   <a href="#">HTTP</a>   <a href="#">PING</a>   ... |
| 甘肃兰州移动 | 211.139.80.6  | 265.21ms  | uexw: [■■■■■] run | -      | <a href="#">详情</a>   <a href="#">HTTP</a>   <a href="#">PING</a>   ... |
| 陕西榆林移动 | 211.137.130.3 | 120.405ms | uexw: [■■■■■] run | -      | <a href="#">详情</a>   <a href="#">HTTP</a>   <a href="#">PING</a>   ... |
| 贵州贵阳电信 | 202.98.192.67 | 46.254ms  | uexw: [■■■■■] run | -      | <a href="#">详情</a>   <a href="#">HTTP</a>   <a href="#">PING</a>   ... |

再回到 Devbox 中配置自定义域名，点击确认。



最后，部署应用。

**基础配置**

应用名称: devbox-release

镜像源: 公共

镜像名: hub.hzh.sealos.run/ns-hj0m6m8e/devbox:v1.0.0

部署模式: 固定实例

实例数: 1

CPU: 0.1, 0.2, 0.5, 1, 2, 3, 4, 8 (Core)

内存: 64 M, 128 M, 256 M, 512 M, 1 G, 2 G, 4 G, 8 G, 16 G

**部署应用**

出现 **running** 的字眼，则代表运行成功。

The screenshot shows the Seafos DevBox interface for the project 'devbox-release'. On the left, the deployment configuration is listed, including basic information like creation time (2024-10-27 16:14), image name (hub.hzh.sealos.run/ns-hj0m6m8e/devbox-release), and resource limits (2 Core CPU, 4 Gi Memory). The network configuration shows an internal address (http://devbox-release-1) and a public address (https://chat.aivideos-sora.com). The instance list shows one running instance named 'devbox-release-1' with 0 restarts and a 3-second start time. A red arrow points to the 'running' status indicator.

如果出现 **error**, 可以点击右侧日志按钮进行查看。

This screenshot shows the real-time monitoring section of the Seafos DevBox interface. It includes CPU usage (0%) and memory usage (1.26%). The network configuration section shows the same internal and public addresses. The instance list shows the single running instance with its detailed metrics. A red arrow points to the log icon in the operations column for the instance row.

最后, 使用自己的域名来访问线上的页面, 成功。

抛开代码编程和调试, 仅针对部署过程, 基本上只需花费 3 分钟左右。

用户: 你是什么大模型

AI: 我是一个大型语言模型,由清华大学KEG实验室和智谱AI公司于2023年共同训练推出。我的任务是针对用户的问题和要求提供适当的答复和支持。由于我是个计算机程序,所以我没有自己的情感、意愿或主观意识,只是一种工具,用于帮助用户解决问题和获取信息。我的优势在于可以处理各种语言任务,并提供专业、准确、有洞察力的建议。

输入您的问题...

发送

## 四、总结

看完是不是发现，原来部署也可以这么简单？

从前我们总觉得，不会编程就做不了产品。可现在不一样了。

我们不需要成为全能选手，  
不需要精通所有技术细节，  
更不需要被那些专业术语所困扰。

技术在进步，工具在迭代。

**Cursor 帮我们写代码，Sealos 帮我们一键部署，曾经让人头疼的技术环节，现在都变得如此简单。**

就算是作为程序员的我，每次用这套组合来做技术验证、快速实现想法，都会感叹：要是早几年有这些工具就好了，能少走多少弯路啊！

如果你正在用 Cursor 开发自己的项目，别被那些技术障碍吓退。

按照今天的方法试试看，你也能轻松把创意变成一个真实的产品，让所有人都能用上你的作品。



【学习安排】跳转入口： [学习安排](#)

◆ **运营有话说**: 学习到这里，对于编程小白的圈友来说已经很棒啦，第七章的内容适合做过网页、能解决外部上网问题 的圈友。很多常规的简单操作太过简单还会占篇幅，就不会事无巨细地记录在手册上。

学习有余力的圈友可以开始做大作业啦：[大作业说明](#)

## AI破局

# 七、建设网站 @小潘

手册里讲了很多AI编程的简单案例，这对于一些原本有编程基础的圈友来说，小菜一碟。本章加深一点难度，讲讲做网页！

做网页难不难呢？

不难！网上找些教程，学上一两个小时，大部份人都能上手。

做高颜值，漂亮精致的网页难不难呢？

很难！我们要懂的东西太多了，要学前端技术，例如HTML、CSS等。但，最关键的还是我们要懂审美！

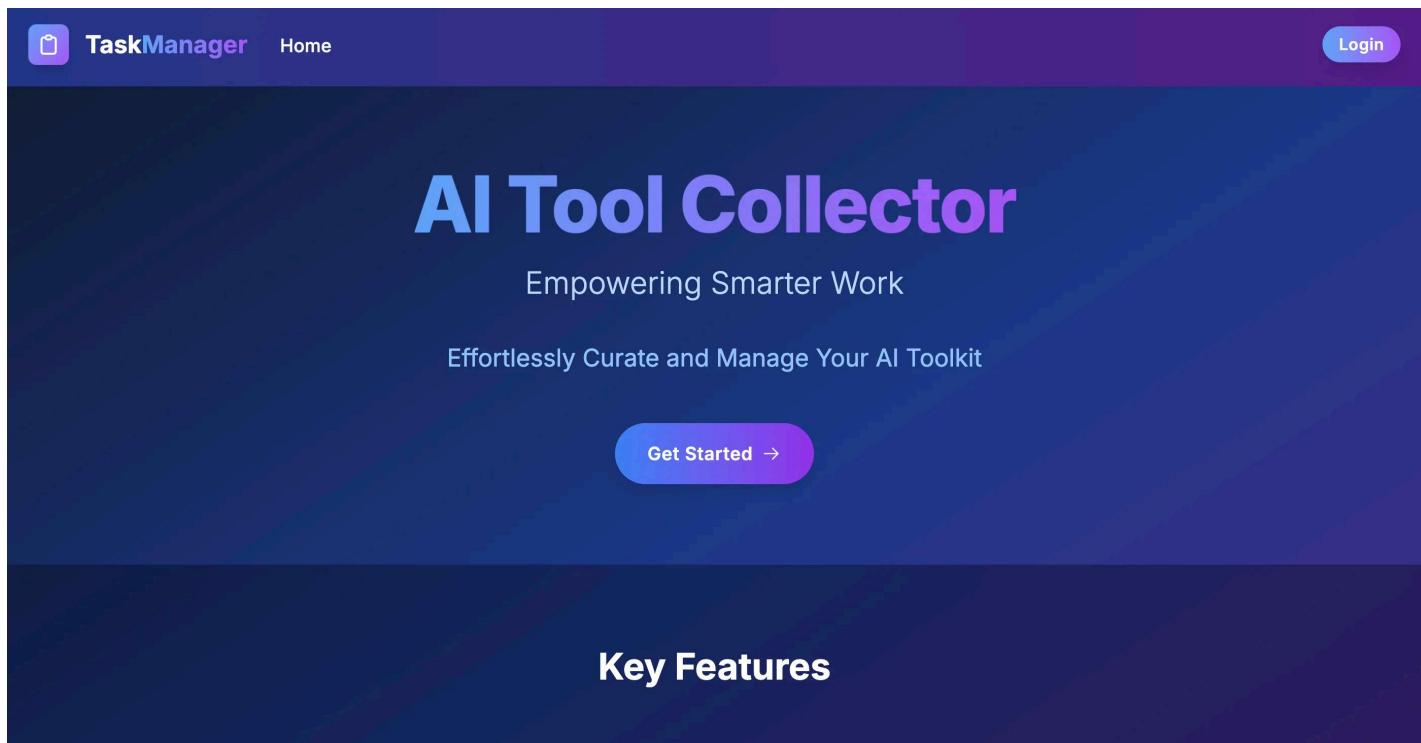
不过，有了Claude、Cursor，不懂代码，不懂审美，做个高颜值的网站，也能像喝水一样简单。

## 7.1 审美炸裂的Claude

ChatGPT也能做网页，我在我的教程里就有详细介绍。但是ChatGPT做的网页风格很单一，离美观还有不小的差距。

Claude3.5的发布后，就有圈友说很牛，用来写网页效果非常炸裂！听到圈友们这么推崇，我也去体验了下，结果没让我失望！

下面这个就是我用Claude生成的网页，整个过程中，我没有写一行代码！



我的提示词非常简单

1 请帮我生成网站的首页，要求没有任何配图，要有科技感和高技感

让我惊艳的点有两个：

### 1)出色的审美能力

我在提示词中和审美相关的有两个方面，科技感和高技感。

那么，网站要如何体现科技感和高技感呢？

网站的配色和动态效果！

Claude为整个网页选择了蓝色和紫色渐变色作为主题风格，另外，打开网页时候还有恰到好处的动态效果。

通过这两点，让整个网页看起来美观了不少。

### 2) 精细化调节能力

在第一轮生成网页的时候，不是一开始就很满意，可能某个文案不好看或者某个按钮样式太丑了。

以前我用ChatGPT时，碰到这种情况，只能手动修改。ChatGPT微调网页的能力并不出色，甚至可以说是难用。

Claude在这方面，可以说完虐ChatGPT，Claude几乎是指哪打哪，就算有错，错误的原因大部分是我们描述不到位。

到这里，小伙伴们应该也知道了，Claude在生成网页方面真的很强，那么，接下来，我会介绍下如何使用Claude生成网页！

## 7.2 零基础使用Claude生成高颜值网页

在开始介绍介绍怎么用Claude生成漂亮好看的网站前，先给大伙儿介绍下我们可以用的工具

### Claude网页生成工具

官方站点，不必过多解释。官方，就是质量的保障。

如果只是用于生成网页，我不建议大伙儿订阅Claude的高级会员。

Claude的高级会员要20\$一个月，但即便是付费的会员，使用Claude的次数也是受限的。反而是一些第三方工具使用Claude并不受限。

下面我就介绍些我知道的，可以立即用起来的工具。

cursor - AI写代码利器（本次行动营主推）

强力推荐！强力推荐！cursor支持claude、gpt等多种AI大模型。

没有任何吹嘘的成分，有了cursor，只要你会打字，零基础也能写出高水平的漂亮网页。



# The AI Code Editor

Built to make you extraordinarily productive,  
Cursor is the best way to code with AI.

[Download for Free](#)
[Watch Demo  
1 Minute](#)

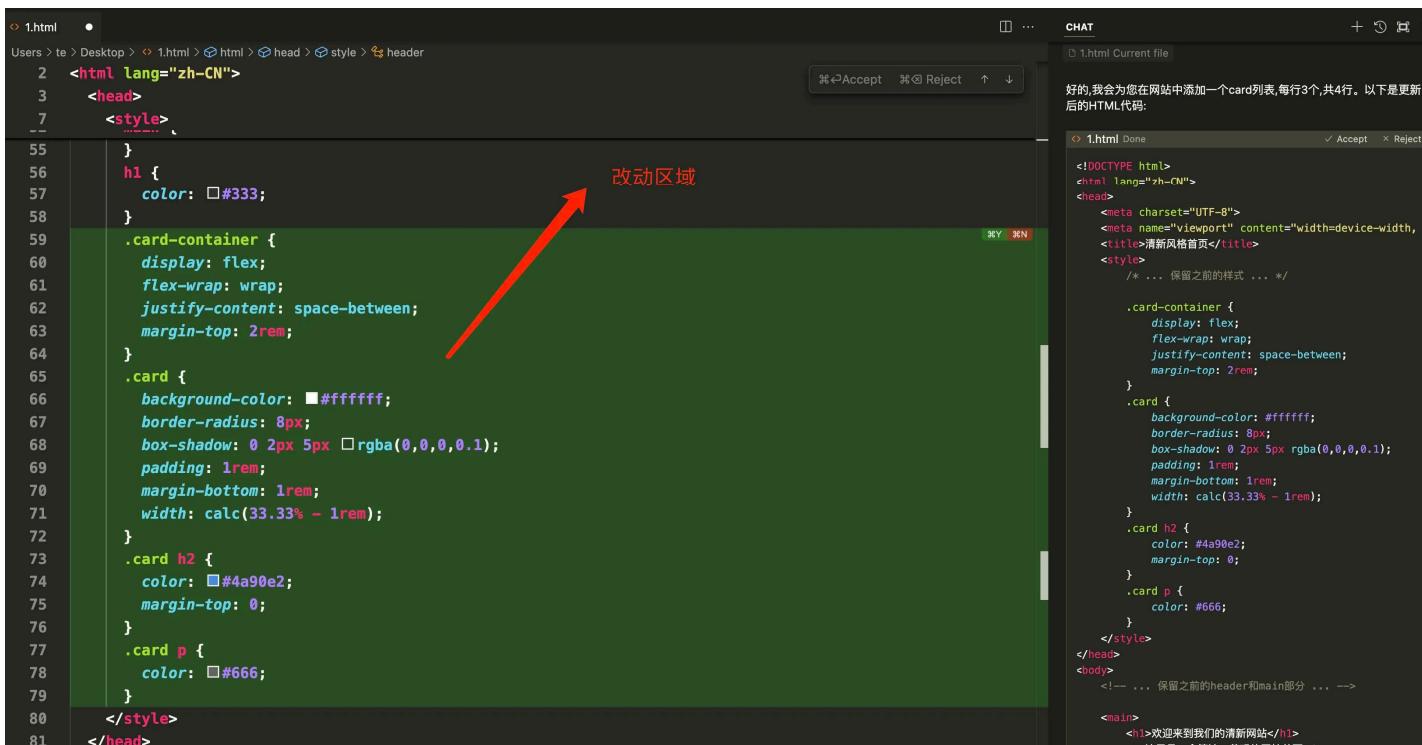

cursor是一款AI代码编辑器，类似于github的copilot，从我的实际体验看，cursor远胜github copilot。

cursor可以让AI扫描整个项目的代码，这项能力得益于大模型的支持的上下文越来越长。相对的，github copilot只能读当前的文件。

另外，我们可以在编辑器内，通过语言描述功能，cursor会精准的生成代码。点击接受生成的代码后，会自动显示在左侧的编辑区。

最让我拍手叫绝的是，cursor非常智能，当我要新增内容时，cursor只会修改新增的部分。而它的老对手GPT一般会全部都改了，因此我们用GPT微调网页极度困难。

另外新增的功能会友好地展示在右侧，并且高亮标识出来。



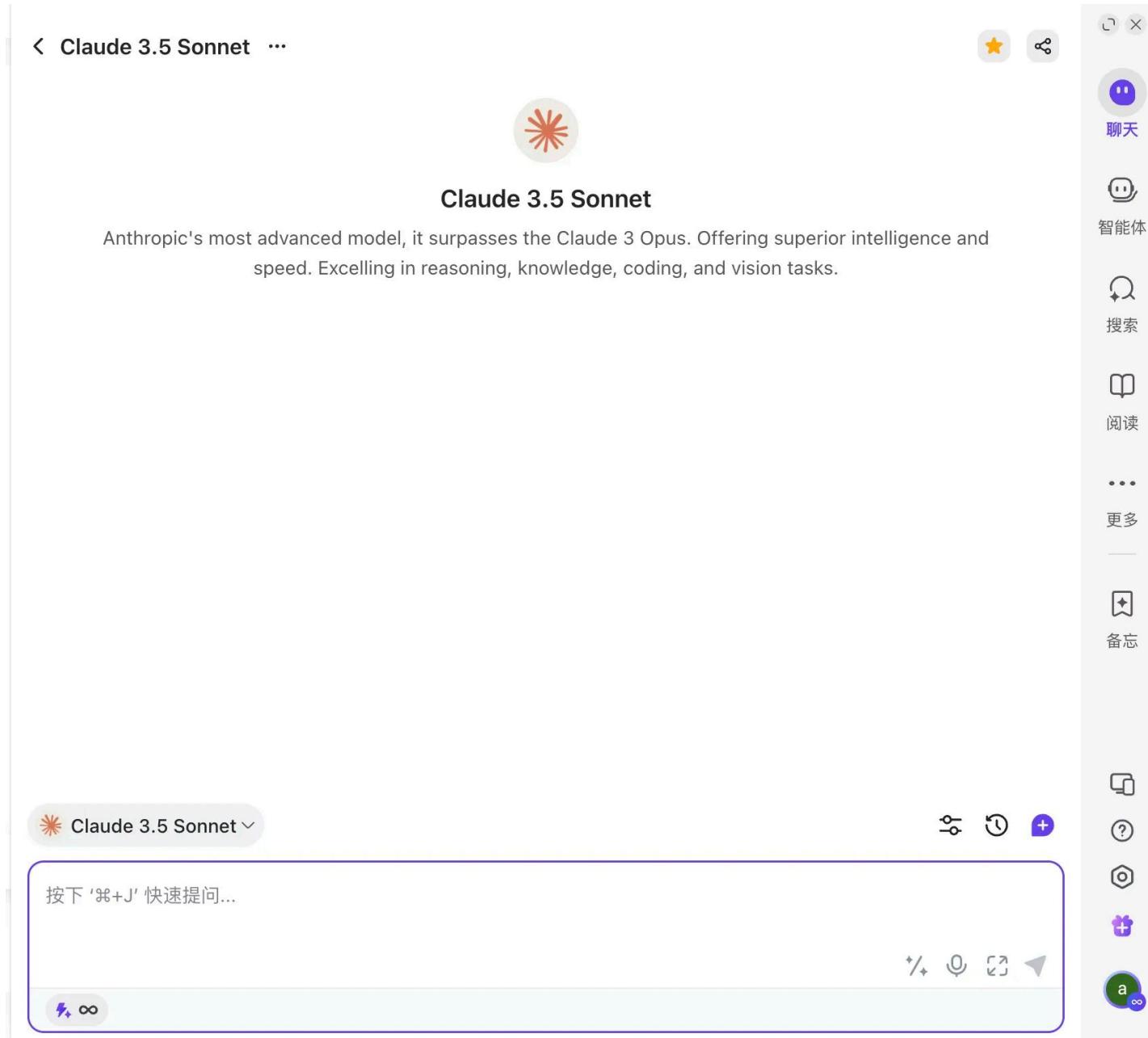
那么，cursor一个月多少钱呢？20\$！确实有点贵，价格是github copilot的两倍，但是，人家cursor确实好用啊。

当然，我那么强力推荐cursor，并不只是他好用，最关键的是，cursor新用户14天免费使用！我们每个人都有机会白嫖14天，14天过期后，再注册一个帐号，又是14天。。。

当然，除了Cursor、Claude外还有其他AI编程的工具，大家都可以尝试一下

monica - all in one的AI助手

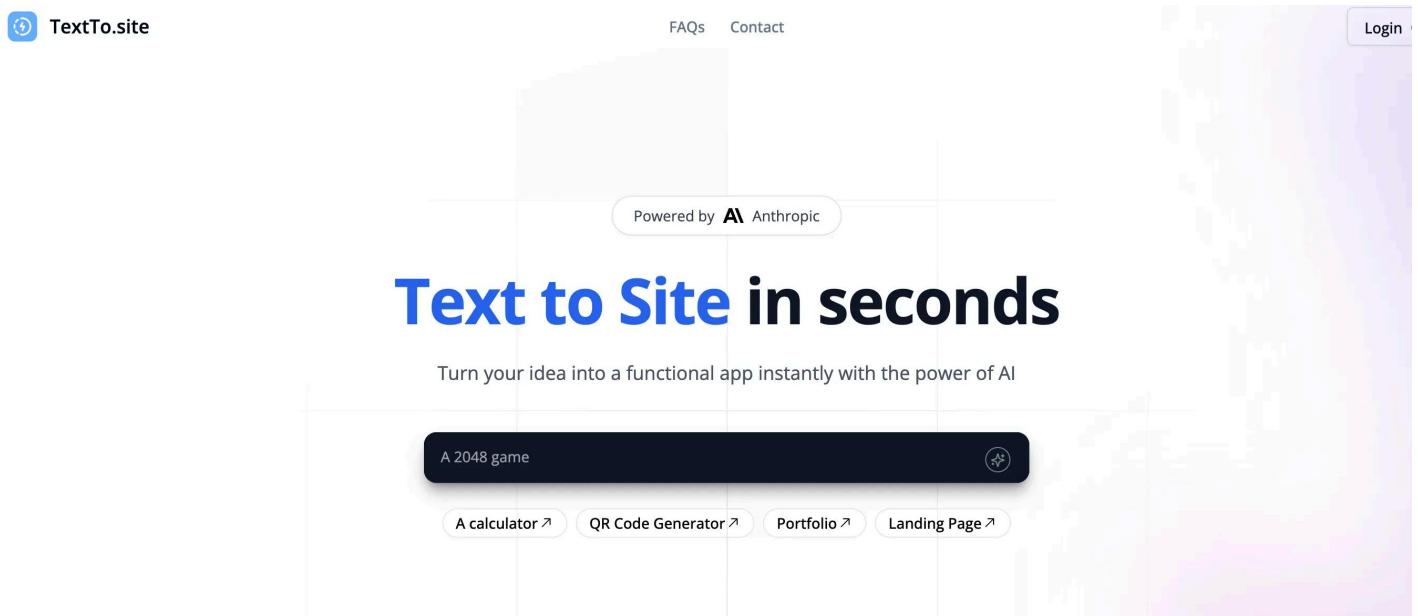
套了各种壳的AI助手很多，但monica是我唯一长期使用的AI助手。



monica也支持了claude 3.5 sonnet模型，只要是付费会员，就可以无限使用claude 3.5 sonnect模型了。

**texto.size**

这个是群友今天新鲜出炉网页生成站点，也是基于Claude。



优点很明显，可以白嫖。缺点呢，就是不支持站点生成内容的微调。

毕竟只是独立开发者，生成网页Claude的费用可不小，生成一个网页估计得几毛钱。

希望站长能够长期坚持下去吧~

传送门:<https://textto.site>

## 7.3 AI生成flux图片生成站点

介绍完工具，接下来以一个小需求为例，讲讲如何使用claude模型来做漂亮的网页。

最近flux很火的，我们就来构建一个flux图片生成站点吧！

**编程小知识：**FLUX.1 是一个全新的 AI 图像生成模型，在图像细节、指令遵从性、风格多样性和场景复杂性等方面，为文本到图像合成定义了新的最先进水平(state-of-the-art, SOTA)。这个模型在开源**文生图模型**领域引起了广泛关注，被认为有可能取代 Stable Diffusion 成为新的领头羊。

### 7.3.1 需求分析

在开始前，我先来简单分析下，一个简单的flux图片生成站点由哪些网页构成呢？

一般来说，一个完整的工具站，大部分都会有下面几个组成部分：

- 1、首页，介绍站点的功能
- 2、功能区，一般是工具界面
- 3、价格介绍
- 4、登陆页面

除了以上主要的页面外，其实还有其他的页面，就不全部列出来了

为了演示方便，我们就只做1个页面，**功能页面和价格页**

🎁 我们先简单讲解cursor的用法，以及如何从零开始创建一个可以被全世界人民访问的网页，以便于后面更好的分享。

#### 项目创建主要分为2部分：

- 1、在github创建帐号同时创建项目，用于管理我们的代码
- 2、在vercel中部署我们在github中的项目，让全世界人都能访问

### 7.3.2 在github中创建项目

github是个代码管理平台，全世界的程序员都会使用。

如果你还没有github帐号，就需要注册一下。关于注册github，实在搞不定，谷歌或者百度搜索「github注册」，都能找到很好的教程。

我们注册好github帐号后，进入：<https://github.com/>



应该能看到图片中「New」按钮，点击New按钮，就能跳转到我们的项目创建页面。



输入项目名称后，将网页拖到最下方，点击「Create repository」就可以完成我们的项目创建了。

完成项目创建之后，会弹出下面的这个页面



代表着我们的github项目已经创建成功了。

不过，这还没有结束，我们需要把这个项目下载到本地，方便我们让AI进行修改。



只需要点击图片中的复制按钮，如果是苹果电脑的话，打开一个terminal的窗口，然后在窗口中执行下面的指令即可。

```
1 git clone 「你的项目地址」
```

执行后，可看到类似下方的效果。



如果是windows电脑，操作也差不多。

完成了项目下载到本地后，打开git clone下来的空目录，创建文件名为index.html的文件，把下方的代码拷贝到目录中。

```
1 <html>
2 <title>这是一个测试页面</title>
3 <body>
4 <h1>
5 这是一个测试页面
6 </h1>
7 </body>
8 </html>
```

具体可参考下方这张图。



其中：

flux-img是我们从github上下载下来的项目

index.html是我们新建的文件

index.html中的内容则是上面的代码。

好了，最后，我们在执行一个命令，把这个项目提交到github上

```
1 git add .
```

```
1 git commit -m "init"
```

```
1 git push origin master
```

依次执行下面三个指令后，我们新增的index.html就会被更新到github项目中。

至此，项目创建，更新项目都完成了。

### 7.3.3 在vercel中部署

(vercel需要外部上网条件，一般到淘宝找。如果实在解决不了，可以跳过这一部分)

这一部分将演示如何在vercel中部署我们在github中的项目，让全世界人都能访问

想让我们的网页挂在网上，被全世界的人看到，有很多方式，我在这里儿使用的是vercel。

vercel是可以免费搭建网站的平台，我们今天所有的演示，都是在vercel进行的。可以通过下面的地址到vercel的官网：[vercel.com](https://vercel.com)

进入官网后，应该就能看到下面的界面了。



点击「Start By Deploying」，就会跳转到一个新的页面。

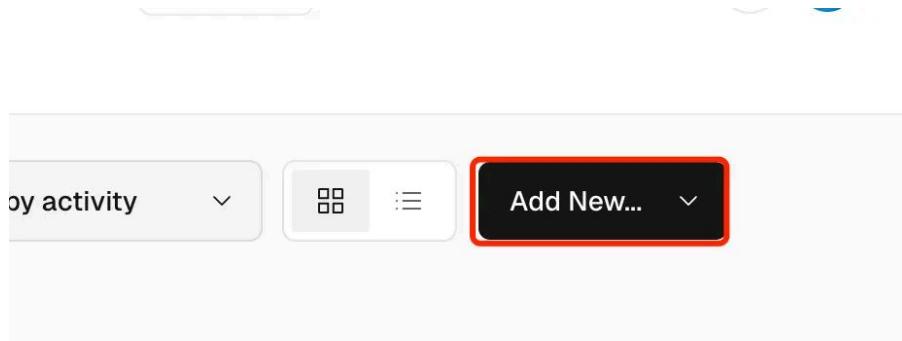


这个新的页面，要求我们先进行登陆。

这个页面支持3种登陆方式，我在这儿选择的是第一种登陆方式「Continue with github」

登陆时，按照界面提示，可以和我们上一步注册的github账号关联。

完成登录后，在Overview的tab下



然后，可以看到，vercel跳转到了创建页面，如下图所示：

## Let's build something new.

[Collaborate on a Pro Trial](#)

To deploy a new Project, import an existing Git Repository or get started with one of our Templates.

这里可以看到，我们新创建的flux-img项目就在列表中，点击import，然后，在弹出的新页面，点击「Deploy」

### You're almost done.

Please follow the steps to configure your Project and deploy it.

经过一小会儿的等待后，就能看到vercel有弹出了一个撒花页面

这表示，我们的项目部署成功了。点击最左侧的小窗，就可以看到我们的网页了

比如，我们这个演示的项目，只要访问（需要外部上网条件）：<https://flux-img.vercel.app/>

至此，如何创建一个项目，如何部署一个项目就完成了。没有操作过的圈友们可能觉得流程多，复杂。实际上，只要熟悉了，上面的这些步骤，全部操作完成也就只需要几分钟。

接下来，就是我们的重点了，怎么让AI帮我们生成网页！

### 7.3.4 让AI帮我们生成网页

我们在此处使用的AI代码编写软件是cursor。到这一章还没有cursor软件的小伙伴，跳回到【[第三章 AI编程工具——Cursor](#)】一步步把工具准备好。

接下来的操作演示是在苹果电脑上进行的，Windows的操作也差不多。

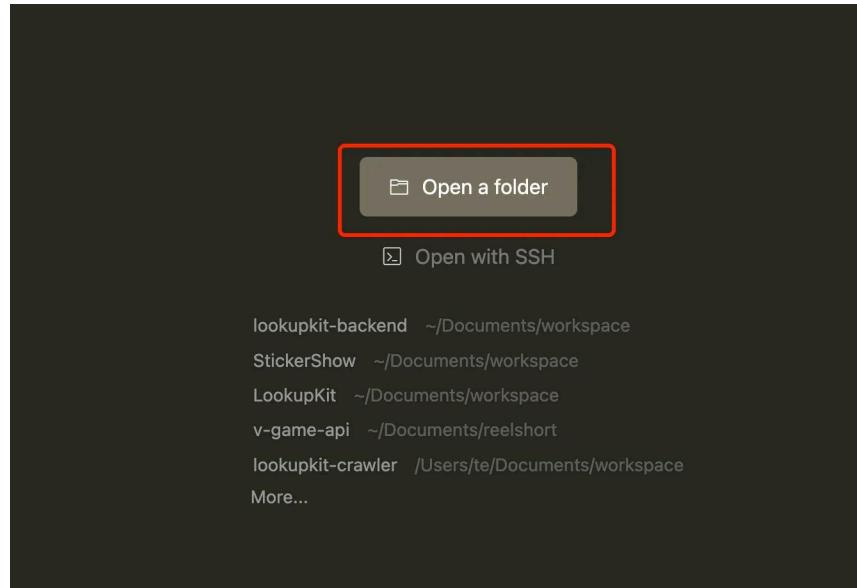
#### 0. 简单回顾Cursor的使用

在下载cursor后，苹果电脑上，应该会看到下面的图标

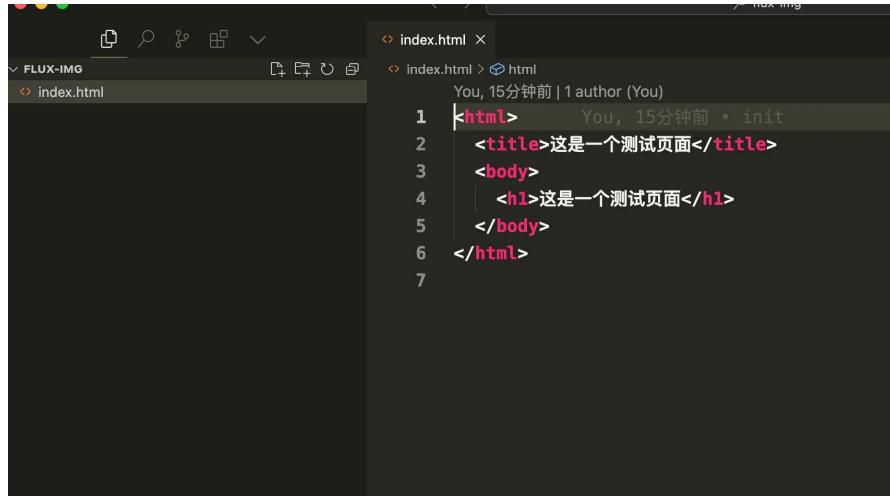


点击后，就可以打开我们的cursor页面了（可能需要预先安装vs code，百度查找如何安装即可）。

点击「Open a folder」，选定上一步中下载到本地的项目



打开后，可以看到类似下面的内容。

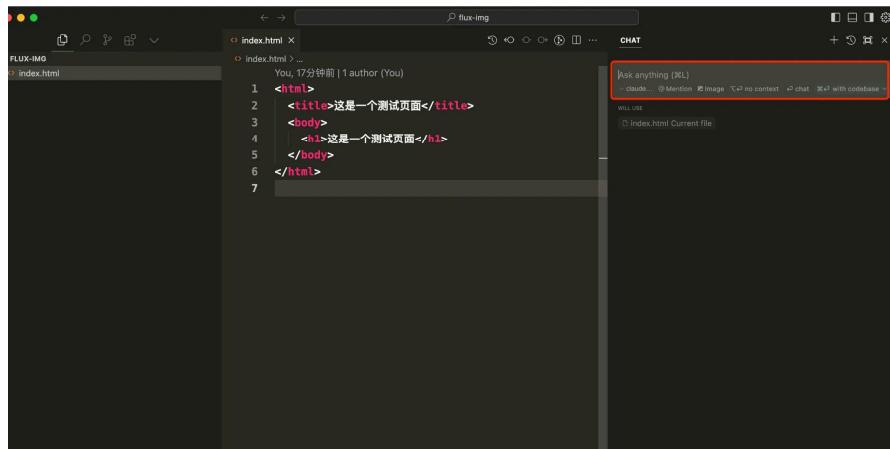


```

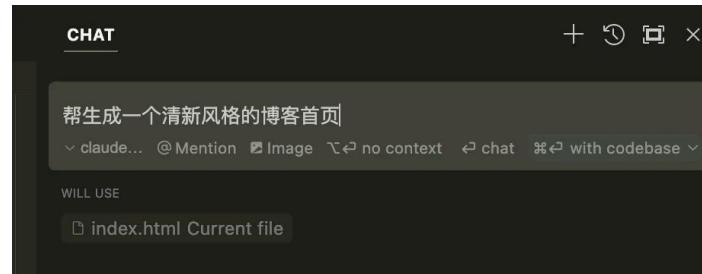
<html>
<title>这是一个测试页面</title>
<body>
 <h1>这是一个测试页面</h1>
</body>
</html>

```

苹果电脑(windows电脑快捷键是「Ctrl + L」), 通过「command + L」可以快捷打开cursor的对话区,如图所示:



在对话区, 可通过语言, 让cursor帮我们生成页面, 例如:



指令执行完成后, 可以看到新页面的代码就在下方了

The screenshot shows a Feishu AI interface for generating a blog homepage. At the top, there's a 'CHAT' tab and a message: '帮生成一个清新风格的博客首页'. Below it, a 'USED' section shows 'index.html Current file'. A code editor window displays the following HTML and CSS code:

```

<!DOCTYPE html>
<html lang="zh-CN">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width,
 <title>我的清新博客</title>
 <style>
 body {
 font-family: 'Arial', sans-serif;
 line-height: 1.6;
 color: #333;
 max-width: 800px;
 margin: 0 auto;
 padding: 20px;
 background-color: #f0f8ff;
 }
 header {
 text-align: center;
 padding: 20px 0;
 background-color: #e6f3ff;
 border-radius: 10px;
 }
 h1 {
 color: #4a90e2;
 }
 .post {
 background-color: white;
 padding: 20px;
 }
 </style>

```

A red box highlights the 'Apply' button at the top right of the code editor.

点击Apply，cursor会自动将代码更新到index.html中。

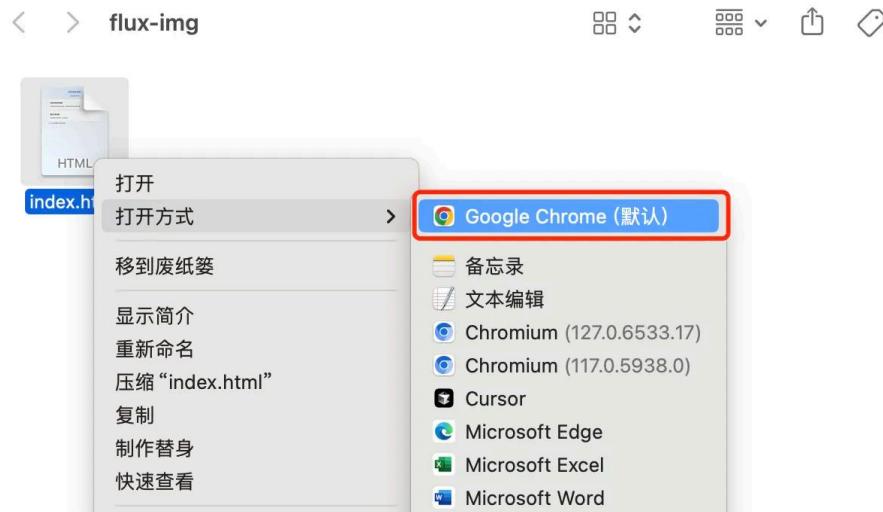
这时候，cursor还需要我们进行二次确认。

This screenshot shows the same Feishu AI interface after the 'Apply' button was clicked. A modal dialog is overlaid on the code editor, containing the generated HTML code and a message: '好的,我来帮您生成一个清新风格的博客首页。以下是修改后的HTML代码,添加了一些基本的结构和样式:'. It includes two buttons: 'Accept' (marked with a checkmark) and 'Reject'.

左侧的代码区中，需要改动的代码和原来的代码会进行高亮。

如果接受改动的代码，只需要点击右侧的「Accept」即可，不需要则点击「Reject」

在Accept后，找到原来index.html的目录，点击，用谷歌浏览器打开，就可以看到效果了。



在我这儿，生成的效果就是下面这样子的了。



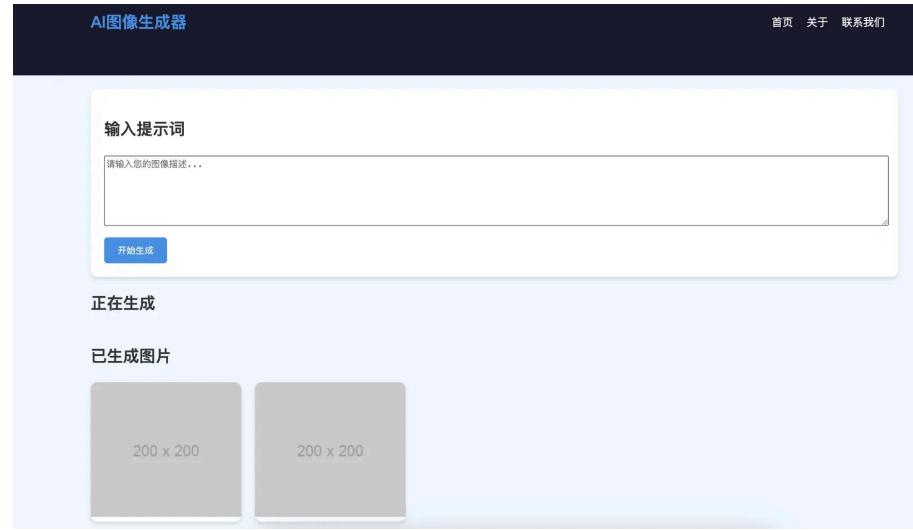
这是个简单的博客页面，但是，这页面如果让前端从零开始写，也得花不少时间，而用Cursor只花了不到1分钟。

好了，cursor的威力给大家见识了，接下来，就开始生成我们的flux-img的首页吧。

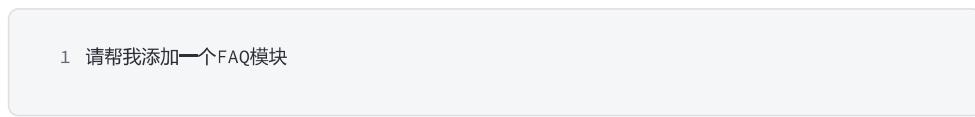
## 1. 生成功能页面

1 我要生成一个文本生成图片工具页面。要求有个提示词输入区，已生成图片列表展示区，正在生成图片展示区。

cursor生成的网页效果如下👉



看起来提到的要素都有了，整体也还行，我们在添加一个FAQ模块。

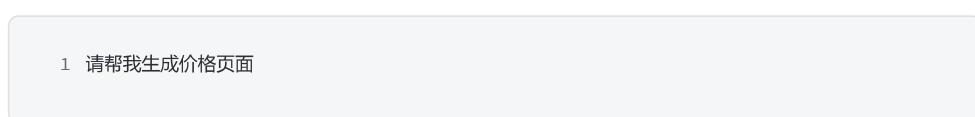


收到指令了，cursor马上生成了下面的模块了，上面的文案，其实也是cursor生成的。



## 生成价格页面

有了前面功能页面生成的经验后，在生成价格页面应该是很简单的事情。



不到一会儿的功夫，就可以看到下面的价格页的效果了



## 2. 页面美化--风格调整

前面的两个页面很快生成了，不过，页面看起来不咋美观啊。

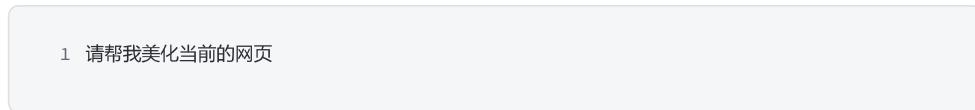
我前面说过，Claude3.5是懂得审美的，cursor是基于Claude3.5的，所以现在生成的页面，还不是cursor的真正实力。

我们还可以再进一步美化。

在指令区，我们先输入 @,然后选定需要调整的页面，比如下面，我们需要调整index.html



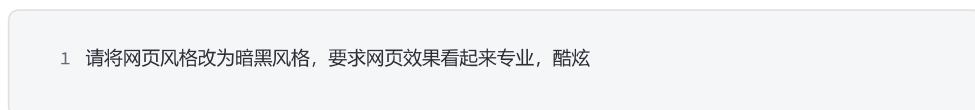
我们可以先不用指定任何风格，先让AI帮我们进行一轮美化

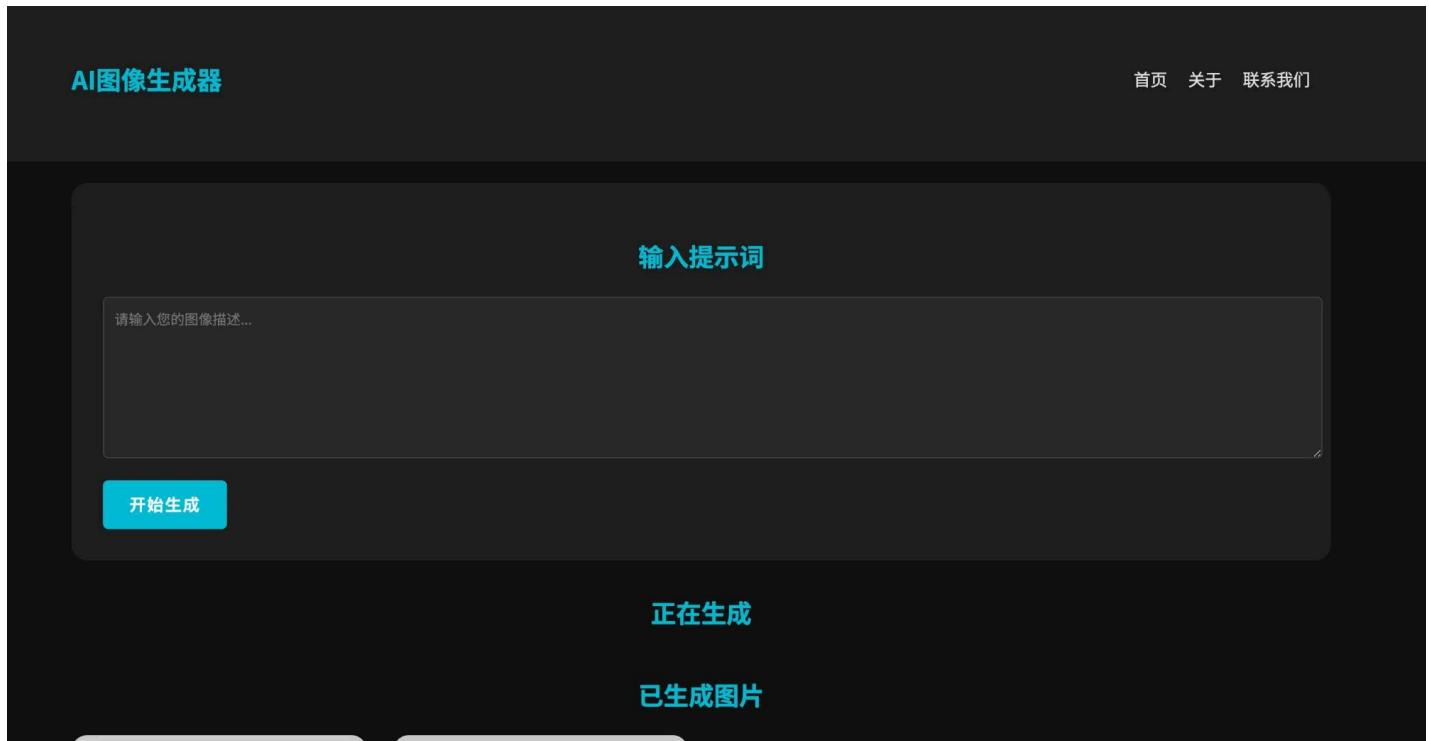


经过一轮美化后，网页经过了席位的调整，相对于上一个版本，稍微好了一些。



我们再进一步，要求对网页的风格进行修改



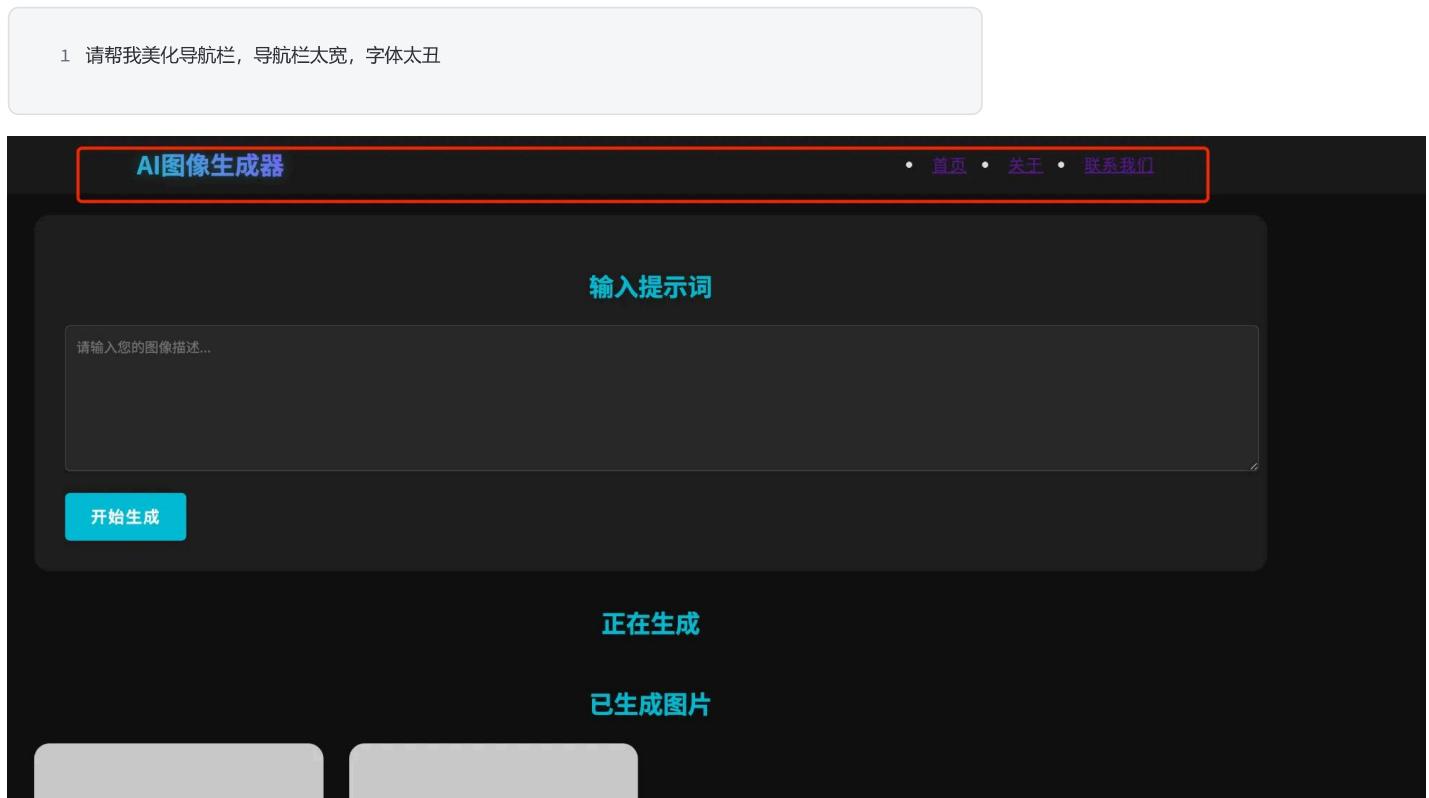


整体风格已经有了很大改观，不过，细节上，还是太粗糙了。

比如，导航栏太宽了，字体也不好看，输入框不够突出。。。等等

我们接下来，针对每一部分都进行优化

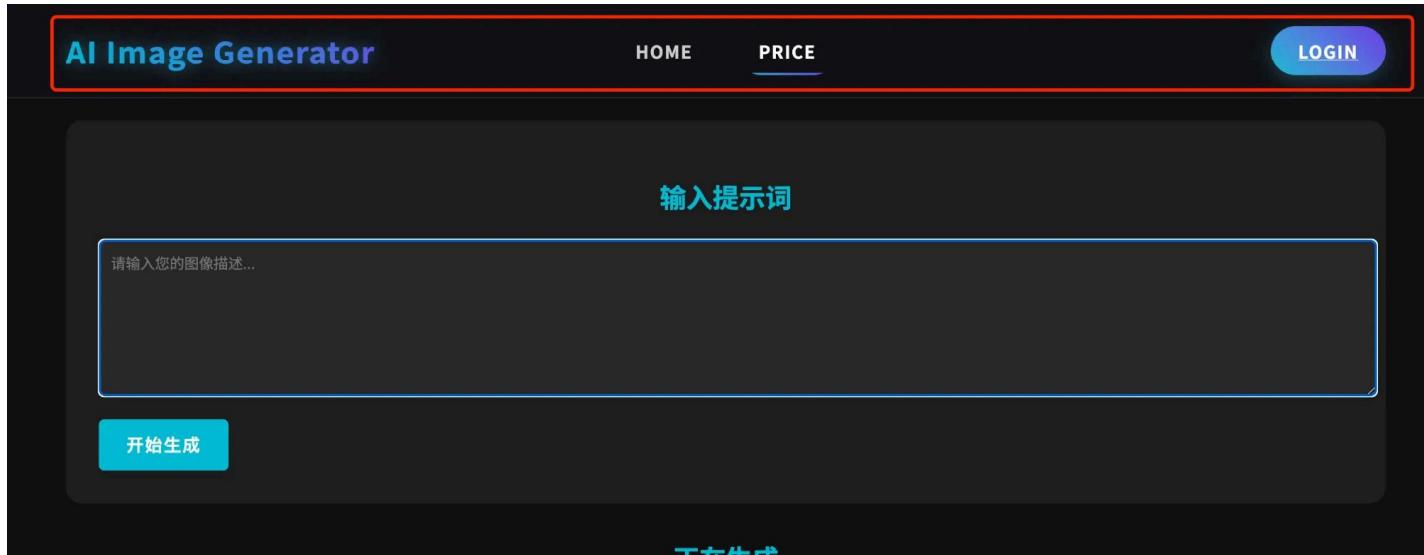
### 3. 细节美化--导航栏



初步调整后，效果还是不够好，我们再进一步调整

1 导航栏的文案改成英文，右侧的链接只有 Login，中间的链接有“Home、Price按钮”，要求链接要突出，方便看，并且要美观

经过多轮的调整后，最终达到了下面的效果



这个效果我还是比较满意的，整个过程的提示词如下

1 AI Image Generator 靠左侧，Home和Price要在导航栏中间，Home去掉按钮背景色

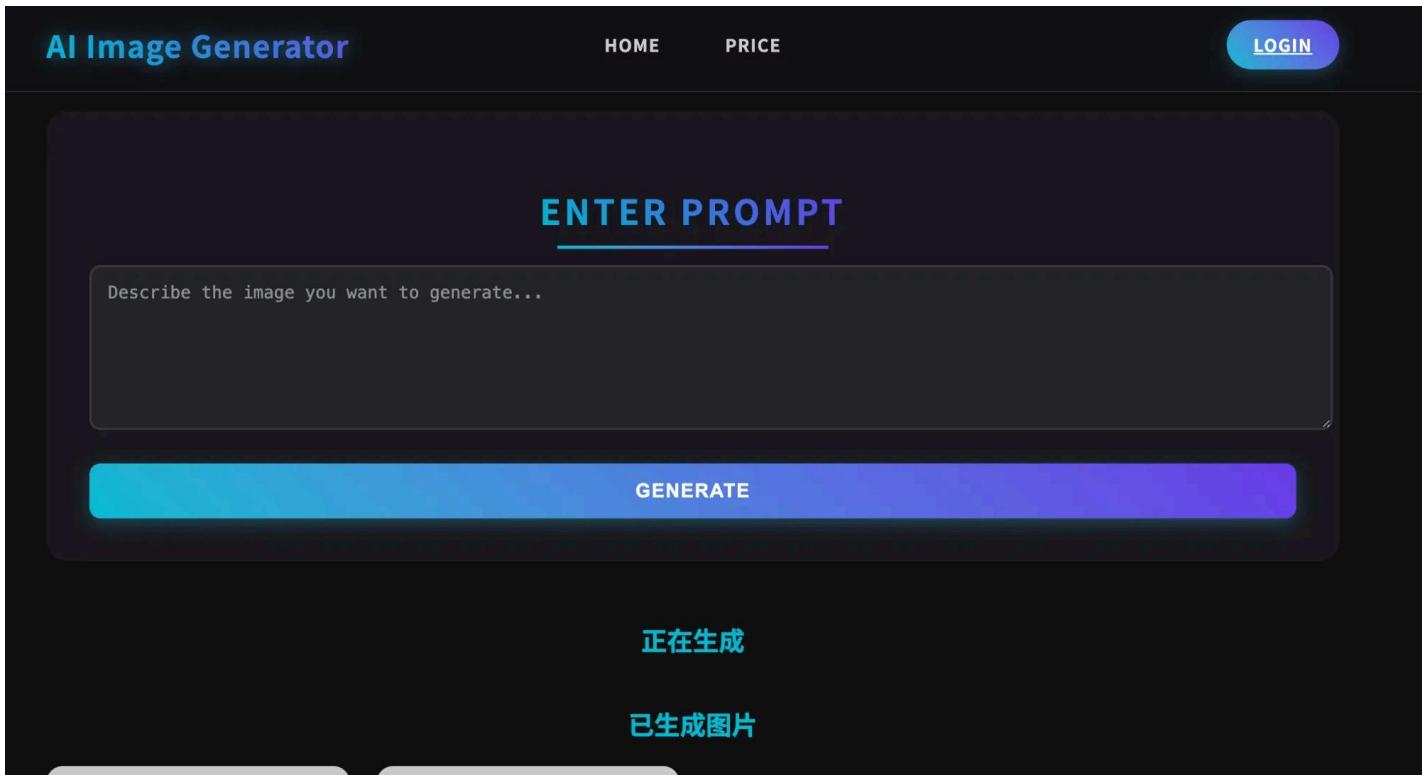
1 Home和Price在导航栏中间

1 美化Home和Price样式

#### 4. 提示词输入模块美化

1 请美化输入提示词模块

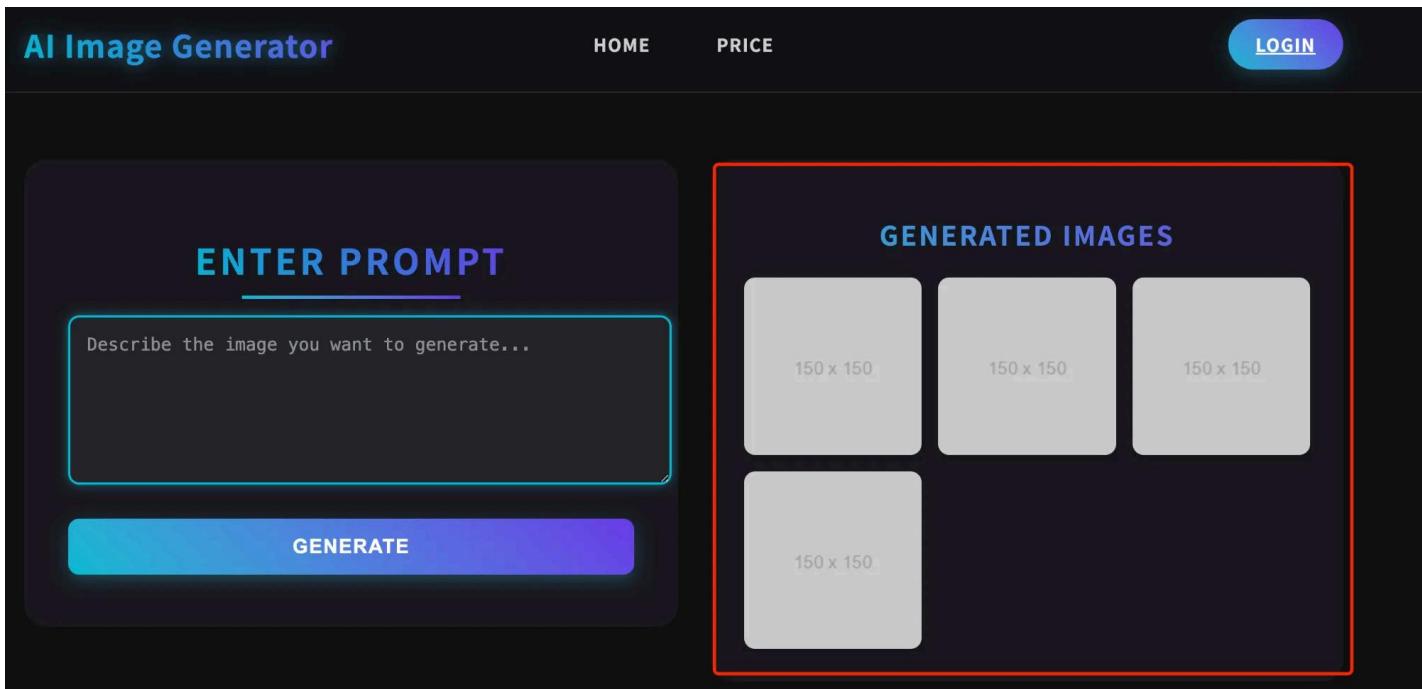
1 输入提示词模块文案改成英文，同时，美化标题



经过两次调整，提示词输入模块看起来更加美观了。

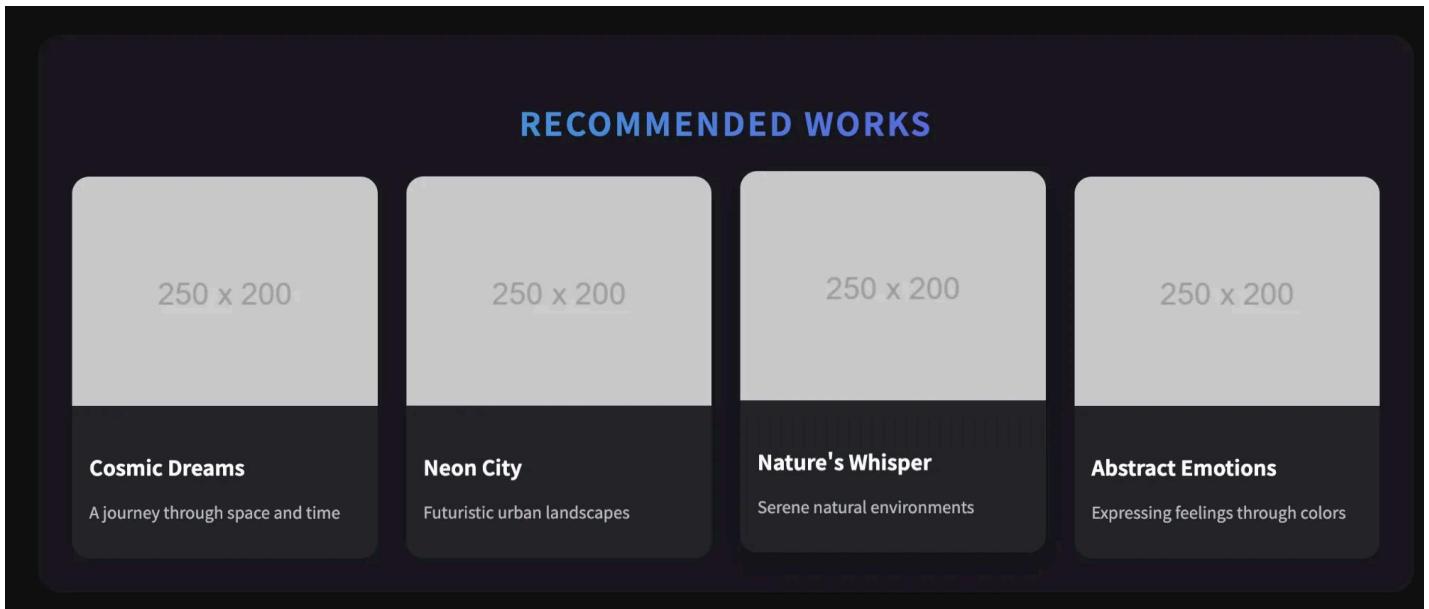
但是，我感觉一个提示词输入框就占据了整个页面的宽度，看起来并不美观，所以，我调整了一下布局

1 提示词输入模块在左边，右边是生成的图片模块



这样子，整个页面看起来更加紧凑了。

接下来，对推荐作品列表



#### FAQ模块

The image shows a dark-themed user interface section titled "FREQUENTLY ASKED QUESTIONS" in blue capital letters at the top center. Below the title are four sections, each starting with a question in blue text followed by a detailed answer in white text.

- How does the AI image generator work?**  
Our AI image generator uses advanced machine learning algorithms to create unique images based on your text descriptions. It analyzes your input and generates images that match the given prompts.
- Can I use the generated images for commercial purposes?**  
Yes, all images generated by our AI are royalty-free and can be used for both personal and commercial purposes. However, we recommend checking our terms of service for any specific limitations.
- How long does it take to generate an image?**  
The generation process typically takes between 10 to 30 seconds, depending on the complexity of the prompt and current server load. More detailed or complex prompts may take slightly longer.
- Can I edit or modify the generated images?**

这两个模块的优化，都是使用同样的方式。

在这个过程中，没有手动添加任何的代码。

#### 5. price页面美化

在前面，我主要集中精力在美化index.html。price.html的页面还是很丑的。

在美化price页面前，有个很重要的点大伙儿要知道——网站的风格一致性。

一个网站的网页，如果一个网页是暗黑主题，另外一个网页是明亮的主题，那么看起来就很别扭。

用AI美化Price页面，很重要的一个点，就是，保持两个页面风格要一致。

注意！

在这里，我让cursor学习了index.html的页面风格

```
<!DOCTYPE html>
<html lang="zh-CN">
 <head>
 <meta charset="UTF-8" />
 <meta name="viewport" content="width=device-width,
 <title>AI图像生成器</title>
 <link
 href="https://fonts.googleapis.com/css2?family=Noto+Sans+SC:wght@400;500;700;900;900italic;900wght@400;500;700;900;900italic&display=block"
 rel="stylesheet"
 />
 <style>
 :root {
 --bg-color: #121212;
 }
 </style>
 </head>
 <body>
 <div>
 <h1>AI图像生成器</h1>
 <p>通过AI技术生成高质量的图像</p>
 <form>
 <input type="text" placeholder="输入关键词" />
 <button>生成</button>
 </form>

 </div>
 </body>
</html>
```

@index.html 请学习index.html的布局方式，和风格样式

cursor的智能让我有点震惊，它竟然知道我的意图是吧price.html的风格改成和index.html一样。

好的，我理解您希望我参考index.html的布局和风格样式来改进 price.html。我会根据index.html的设计风格对price.html进行修改。以下是修改后的price.html代码：

```
<!DOCTYPE html>
<html lang="zh-CN">
 <head>
 <meta charset="UTF-8" />
 <meta name="viewport" content="width=device-width,
 <title>产品价格 - AI图像生成器</title>
 <link
 href="https://fonts.googleapis.com/css2?family=Noto+Sans+SC:wght@400;500;700;900;900italic;900wght@400;500;700;900;900italic&display=block"
 rel="stylesheet"
 />
 <style>
 :root {
 --bg-color: #121212;
 }
 </style>
 </head>
 <body>
 <div>
 <h1>AI图像生成器</h1>
 <p>通过AI技术生成高质量的图像</p>
 <form>
 <input type="text" placeholder="输入关键词" />
 <button>生成</button>
 </form>

 </div>
 </body>
</html>
```

最终，cursor没有让我失望，整个price页面经过美化后，看起来非常专业了。

Plan	Price	Generations per day	Resolution	Customer Support
Basic	\$19/month	100	Standard	Basic customer support
Pro	\$49/month	500	High	Priority customer support
Enterprise	\$99/month	Unlimited	Ultra-high	24/7 dedicated support

自此，两个网页我已经用AI完成了。

前面写了一大堆，那么，用AI做完这两个网页花了多长时间呢？

加上调整和美化的话，加起来也没有超过10分钟。

那么，如果让人来做这网页要花多久呢？

按照一个人一天300行代码算，应该也得2天左右。。还没有加上各种UI设计。

### 7.3.5 部署上线

网页做出来，必须得让其他人也访问才有意义，接下来，我们来讲讲如何部署到vercel。

在前面的步骤中，我们已经在github创建了我们的项目了。并且，也将github和vercel关联。

在前面的步骤中，我们已经在github创建了我们的项目了。并且，也将github和vercel关联。

我们只需要把我们的代码提交到github，然后更新下即可

```
1 git add .
```

```
1 git commit -m "add page"
```

```
1 git push origin master
```

按顺序执行指令，那么就可以将代码更新到vercel了。

```

[NEW BRANCH] master > master
→ flux-img git:(master) git add .
→ flux-img git:(master) ✘ git commit -m "add page"
[master ea2bd11] add page
 2 files changed, 865 insertions(+), 3 deletions(-)
 create mode 100644 price.html
→ flux-img git:(master) git push origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 4.15 KiB | 4.15 MiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:rotbit/flux-img.git
 64d590d..ea2bd11 master -> master
→ flux-img git:(master) █

```

提交到vercel后，vercel会自动更新我们的项目。（外部上网环境） <https://flux-img.vercel.app/>

这个链接发给其他人后，他们也是可以直接访问的。

后续，可以绑定自己的域名，这个操作也不难，可以在godaddy等购买域名后，就可以在vercel进行绑定了。然后通过链接，就可以看到我们更改后的项目了。

## 7.4 站点推广（了解一下）

学习了这么多技能，尝试了这么多案例，那做的这些如何变现呢？

这里给大家提供一个思路，出海站点推广。一般有下面的几种方式。

### google search console

<https://search.google.com/>

将你的站点提交GSC后，谷歌会更快收录你的站点，当用户搜索关键词的时候，就机会看到你的网站。

### Hack News

当完成我们的工具后，可以提交到Hack News，Hack News的访问量很大，当我们提交的内容活了之后，有机会获得更多曝光量和外链（外链和谷歌搜索排名有关，一般外链越多越好）

## Hacker News new | threads | past | comments | ask | show | jobs | submit

Please read the [rules](#). You can also browse the [newest](#) Show HNs.

1. ▲ **Show HN: Hacker News for film buffs** ([newyorkdailyinquirer.com](http://newyorkdailyinquirer.com))  
71 points by asolis0105 12 hours ago | 35 comments
2. ▲ **Show HN: Omi – Open-source AI wearable for capturing conversations** ([github.com/basedhardware](http://github.com/basedhardware))  
84 points by kodjima33 12 hours ago | 61 comments
3. ▲ **Show HN: InstantDB – A Modern Firebase** ([github.com/instantdb](http://github.com/instantdb))  
1078 points by nezaj 1 day ago | 276 comments
4. ▲ **Show HN: Ruroco – like port knocking, but better** ([github.com/beac0n](http://github.com/beac0n))  
108 points by mschempp 1 day ago | 98 comments
5. ▲ **Show HN: I built an icon builder to make beautiful icons fast and free** ([expoicon.com](http://expoicon.com))  
2 points by rudolfsrijkurus 4 hours ago | 4 comments
6. ▲ **Show HN: Support.dev – a non-trivial Vue and Supabase app** ([github.com/bwship](http://github.com/bwship))  
20 points by bwship 16 hours ago | discuss
7. ▲ **Show HN: One Dollar Feedback** ([twocents.site](http://twocents.site))  
2 points by artistaiden 5 hours ago | discuss
8. ▲ **Show HN: Monogo – Evolve your go workspace to a real monorepo** ([github.com/nicolasgere](http://github.com/nicolasgere))  
16 points by elmazout 15 hours ago | 1 comment
9. ▲ **Show HN: AI Tool Is Now Supporting React, Angular, CSS, Svelte, Vue** ([webcrumbs.org](http://webcrumbs.org))  
94 points by m4rcxs 20 hours ago | 73 comments
10. ▲ **Show HN: FOSD – Framework of Operating System Development** ([github.com/fosd-project](http://github.com/fosd-project))  
50 points by i70h4n 1 day ago | 6 comments

## Product hunt

新产品必然提交的站点，虽然有不少人在拉票，但是上了featured后，能获得不错的曝光量。

## Top Products Launching Today

[Featured](#) | [All](#)

For the first 4 hours of the day, we're hiding upvotes to help every product get a chance to catch your interest.  
[Read more](#)

-  **My Journey** — Visualize your whole life through a 3D Timeline ↗  
Q 17 • Bootstrapped • Artificial Intelligence • Quantified Self

▲  
-
-  **Griddr. - A Grid Camera App** — Capture perfect compositions with various grids ↗  
Q 10 • Art • Photography • Photo & Video

▲  
-
-  **WP Adminify** — Extensive WordPress admin customization ↗  
Q 19 • Bootstrapped • WordPress

▲  
-
-  **Aurore.ai** — Gaming and working with companion ↗  
Q 9 • Solo maker • Bootstrapped • Artificial Intelligence

▲  
-
-  **Hey!** — AI-Powered Pair Programming Friend! ⚡ ↗  
Q 11 • Solo maker • Developer Tools • Artificial Intelligence

▲  
-

🎁 【学习安排】跳转入口: ⭐ 学习安排

AI破局

## 八、🔥 大作业要求

大作业提交入口: <https://d16rg8unadx.feishu.cn/share/base/form/shrcn7PjmaoXoGSTWBVYCqVkwC>

为检验大家的学习成果，同时激励大家深入掌握和应用所学技能，我们将在本次线上训练营中开展一次“大作业”活动。通过完成一个完整的谷歌插件项目，帮助大家从自己身边挖掘需求，并且将技能转化为实际成果，同时在总结过程中加深对知识的理解与反思。

### 8.1 活动形式

1. **自由组队或独立完成**: 学员可选择组队（3人以内）或个人完成“谷歌插件制作”。
2. **作品提交**: 作品完成后，以复盘文章的形式进行提交到星球，并带上“#AI编程大作业”标签。复盘内容应包含作品设计、作品制作过程、遇到的挑战和解决思路等。

### 8.2 作品要求

1. **作品主题**: 开发一个有实际用途的谷歌插件，主题不限，大家可以自己身边寻找需求。
2. **复盘文章结构** (建议) : 项目背景 - 功能说明 - 制作过程 - 问题与解决方案 - 成果展示 - 未来改进思路。

### 8.3 评审标准

教练团将从以下四个方面进行评分：

- **创新性**: 插件在功能或设计上是否具有独特性和新意。
- **用户体验**: 插件的界面友好度、操作便捷性等。
- **实用性**: 插件是否满足实际需求，有无应用场景。
- **复盘内容**: 复盘贴是否结构清晰，流程详细，内容完整。

### 8.4 作品提交

1. 制作一个谷歌插件，并把项目代码打包保存到网盘
2. 将作品制作流程、思路记录下来，展示插件使用效果，文章开头就放上网盘链接/插件使用入口
3. 在星球上发布帖子，带上文章链接、“#AI编程大作业”标签
4. 把帖子链接、作业相关信息提交到“大作业提交入口”里

**作业提交时间**: 11月11日-11月18日

大作业提交入口: <https://d16rg8unadx.feishu.cn/share/base/form/shrcn7PjmaoXoGSTWBVYCqVkwC>

🎁 【学习安排】跳转入口: ⭐ 学习安排

AI破局