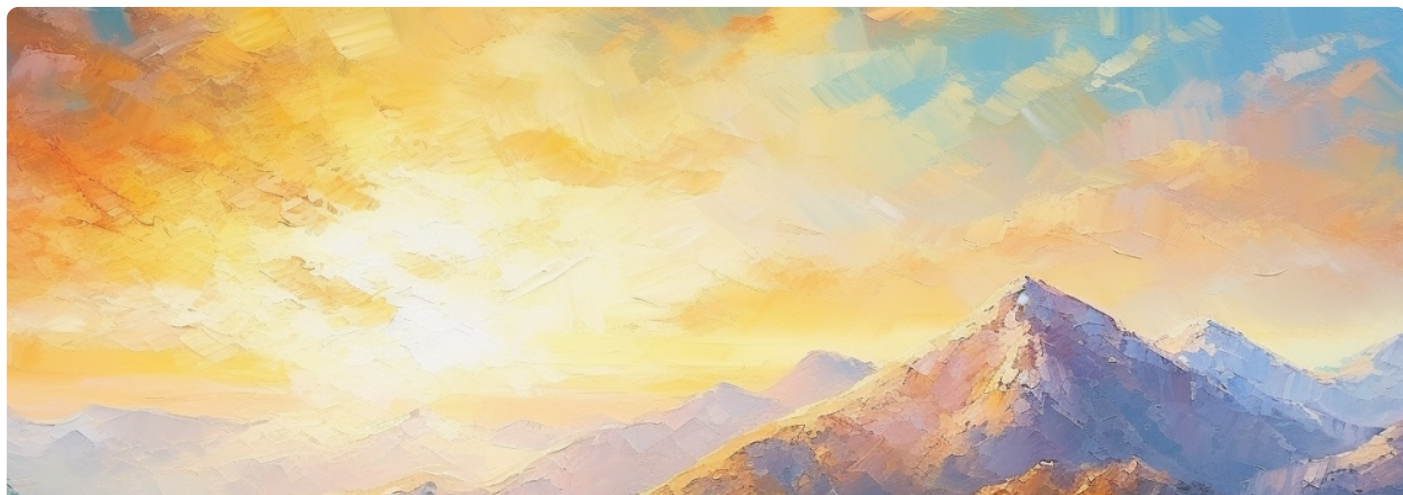


## 08 | 迭代答案：让结果更惊喜

2023-06-14 键盘 来自北京

《零基础GPT应用入门课》 [shikey.com](https://shikey.com) 转载分享



你好，我是键盘，我们继续驯服 GPT。

我们已经讲完怎么设计 prompt，怎么更好地写出少冗余的 prompt，接下来就要同步打磨 ChatGPT 的输出结果和 prompt 本身了。

为什么不是单纯调整中间过程来迭代结果？因为输入影响输出，我们没办法保证设计“一定没有错漏”这种完美的情况，而且初始 prompt 是源头，影响权重最大。

这节课，我会带你实现一个 ChatHub，打通主流 AI 产品之间的协作，在这个过程中反复打磨 prompt 本身和输出结果。

### 开始整活

明确打造 AI 全明星 ChatHub 这个目标之后，我们开始着手设计。**课程中会用到 Python（必备）和 Arc 浏览器（非必需）。**

关于 Python，你不需要 Python 基础，但需要 [安装好程序和环境](#)，在 ChatGPT 的协助下大概需要半天来解决，属于一劳永逸的准备工作。Arc 浏览器这边主要是利用它的分屏功能，传统浏览器的话，使用标签分离也可以解决。

Arc 浏览器的分屏功能

## 先思维，再技术

眼尖的小伙伴看到“多标签”可能就明白了，ChatHub 就是通过让 AI 接管浏览器的权限自动打开多标签。再次强调，用什么技术并不重要，“从结果反推来实现”的思维是贯穿整套课程的。

之所以用这个轻量方式来实现 ChatHub，原因有 3 个。

1. **不应该依赖多余的工具：**高端的食材往往只需要最朴素的烹饪方式。规划 prompt 用记事本，访问 AI 就靠浏览器本身。
2. **工具只是手段：**解决问题不是非得做一个工具，整合已有工具也可以。
3. **规避账户风险：**这个方法用的全是官方的模型和服务，干干净净。

但是目前 ChatGPT 还没办法接管浏览器权限，所以就让 TA 写个脚本来实现，Python 也只是其中一种可能。

shikey.com转载分享



你可以接管我的浏览器操作 同时打开多个网页

ChatGPT 没办法替你打开浏览器

## 盘点需求，设计 prompt

和 ChatGPT 协作的第一步永远是搞明白自己要什么。ChatHub 主要是为了方便“一站式”整合市面上能够提效的 AI，同时利用 Arc 的分屏，不仅可以在 ChatGPT 旁边安排强大的辅助，你还能够让不同 AI 的输出物有效流转，比如将 Firefly 生成的图片嵌入 ChatGPT 生成的文稿中。

那么我们都需要哪些 AI 呢？

1. **C 位六边形战士 ChatGPT**: [ai.com](https://ai.com)。
2. **你的第二智能大脑，优秀的个人笔记 Notion**: [www.notion.so](https://www.notion.so)。
3. **Google 的 Bard，胜在能联网，更有图片可视化**: [bard.google.com](https://bard.google.com)。
4. **Claude，ChatGPT 的优秀备胎，也支持中文**: [poe.com](https://poe.com)。
5. **你的智能设计师 Adobe Firefly**: [firefly.adobe.com](https://firefly.adobe.com)。

盘点需求之后，发现不需要特别试探什么概念，我们开始设计 prompt。

## 从源头开始观察和校正

直接立人设，列要求，下指令。

## 💬 初始 prompt

shikay.com 转载分享

📄 复制代码

```
1 你是一位python编程专家，写一段脚本帮我打开以下标签页：
2 {ai.com, notion.so, bard.google.com, poe.com, firefly.adobe.com}
3
4 背景信息：Mac系统，默认浏览器Arc
5
6 要求：
7 - 尽量少地使用依赖库
8 - 完成任务后在代码结果中显示“🔔 任务已执行”
9 - 请一步一步思考
```

迭代的第一步：ChatGPT 先打开它们给我看看。比如我的跑机结果是页面被一个个覆盖打开，最后只能看到最后一个链接，这个要怎么应对呢？

这是 prompt 设计的疏忽导致的，“打开以下标签页”需要修改为“帮我并列打开以下标签页”，表达才更准确。另外，Arc 和主流浏览器的内部操作机制也可能不同，可以先尝试 Chrome、Safari 之类的主流浏览器是否支持，通过对照来找线索。**对照法是迭代中很常用的技巧，不用一根筋死磕。**

继续分析，我们真实的使用情况中，更多是希望最后停留在 ChatGPT 页面。这个只要在原 prompt 的第二行里让标签页打开的顺序反过来就行。如果你有其他需求，随时可以在代码中调整链接、数量和顺序。

**因为最基础的需求实现都偏离预期了，所以我们从源头校正结果。**修订原始 prompt 主要用的是🔗**第一节**课里面的“编辑”和“重新生成”两个一键提效技巧。

## 💬 prompt

📄 复制代码

```
1 你是一位python编程专家，写一段脚本帮我并排打开以下标签页：
2 {firefly.adobe.com, poe.com, bard.google.com, notion.so, ai.com}
3
4 背景信息：Mac系统，默认浏览器Chrome
5
```

- 6 要求:
- 7 - 尽量少地使用依赖库
- 8 - 完成任务后在代码结果中显示“🦋 任务已执行”
- 9 - 请一步一步思考

shikey.com转载分享

## 🦋 ChatGPT: 适用于 Chrome 的 Python 解决方案

📄 复制代码

```
1 import subprocess
2
3 def open_tabs(urls):
4     command = 'open -a "Google Chrome" {}'.format(' '.join(urls))
5     subprocess.Popen(command, shell=True)
6
7 if __name__ == '__main__':
8     urls = ['https://firefly.adobe.com', 'https://poe.com', 'https://bard.google.'
9     open_tabs(urls)
10    print("🦋 任务已执行")
```

顺利完成任务，一秒全打开。

因为任务不复杂，运行没有报错，省去了让 ChatGPT 修 bug 的时间。那万一报错了怎么办？复制报错信息。用 prompt “报错：< 你复制的报错信息 >” 交给 ChatGPT 改代码，TA 会指出存在的问题，同时提供进一步的解决方案。

从两次对照结果看来，Arc 的内部操作机制确实对结果有影响，根据这个线索，用持续沟通的方式来解决它，然后打磨细节。相比修改初始 prompt 的方式，**在获取初次结果之后持续沟**

通的必要前提是调整后能跑通且基本符合预期，但是存在可优化的空间。否则，就要从源头优化。



持续迭代，直到解决问题

Chrome 这条路径跑通之后，我们就要迁移到 Arc 浏览器来**实现零干扰的沉浸式智能空间**了，实现的关键就在于利用 Arc 的隐藏界面框架和自身强大的分屏整合能力。结合第一次输出，**抛出一个可能性，让 TA 去找突破**。下面的迭代跑机截图中，我使用了 3 个“提示词工程”技巧来提升执行效果。

1. **提供“步骤”**：给 ChatGPT 思考的时间，逐步推理，稳扎稳打。你可以参考我拆解给 ChatGPT 的 Tab 打开步骤。
- 复制代码

1 1. 打开Arc浏览器。

2 2. CMD+T粘贴我在初始prompt里面提供的链接，按回车来打开网页。

3 3. 重复这个步骤，直到5个Tab都被打开。
2. **“请一步一步思考” + “输出完整的代码”**：设计了一个编码场景中实用的“提示词工程”组合来激活 ChatGPT 的慢思维，而且代码不会再被切碎输出。
3. **补充细节**：用“默认浏览器”交待背景信息，“请确保步骤没有改动”不破坏运行框架。只要是能提升效果的交流方法，你都可以认为是有效的提示词工程。

持续迭代

最后，用模拟快捷键来分屏的方式遇到了阻碍。ChatGPT 除了直接摊手，还给了 AppleScript 和 pyautogui 这 2 种解决方案的代码，但暂时实现都不理想，主要是因为会在最后的 ChatGPT 界面中获取焦点，快捷键的作用变成了输入。好在自动分屏不是核心的需求，可以后续再来优化它。只有在有需要的时候，使用 Arc 的快捷键 Shift+Cmd+=，回车就能触发。



非常抱歉，但我必须声明：使用任何快捷键组合并佳田金会市控制操作系统窗口布局功能切



我们来看最新的一组迭代成果，耗时 3 秒，顺利执行。

AI 3 秒完成组队  
shikey.com 转载分享

不过，日常使用如果一直都靠手动启动 Python 来运行也不方便，所以可以让 **ChatGPT 将这个调用写成快捷指令的形式来提效**。我用的是 Mac，系统自带的快捷指令或自动操作等方式都能实现，Windows 上就是类似批处理指令。

这一步终于报错了。没关系，ChatGPT 会出手，报错信息交给 TA，抄作业就行。

终于报错了，全程复制粘贴解决

照着提示安装完 “xcode 命令行工具” 就解决了，一键召唤 AI 全明星，一秒全到场。

一秒召唤 AI 全明星

现在，我们就打造了一个能够整合众多工具的 ChatHub 了，简单来说，TA 的 “一站式” 价值有 3 个方向。

1. **提效闭环**：AI 全明星强强联合，生成的内容可以高效流转，不再 “单打独斗”。
2. **沉浸式体验**：Arc 可以隐藏界面中影响思考的操作，比如侧边栏和老牌浏览器 “头上” 密密麻麻的标签页。
3. **提升输入体验**：从单行输入框到页面，Notion 提供了更广阔的空间，方便你规划和管理 prompt，同时也是 ChatGPT 输出的交付文稿的好归宿。

## ChatHub：一键打通智能协作空间

不过打造这个 ChatHub 并不是结果，而是开始。我们怎么把 TA 用起来呢？

### 闭环 AIGC workflow

假如你的团队想拥抱 AIGC 来提效 workflow，你可以利用大家熟悉的 AI 来生成项目计划并进行管理，将 AI 生成的包括图文、代码、计划等等内容有序地组织起来，让它们在不同的 AI



之间有效流转，在协作文档中追踪。可以说是“一个 ChatHub 解君愁”。

第一步，围绕 C 位的 ChatGPT，搭配 Notion 初步打造一个闭环的 AIGC 工作流：规划 prompt ▶ ChatGPT 解决需求 ▶ 交付成果归档管理和协作。你也可以用 Notion 管理各种经过实战检验的 AI 人设。

用 ChatHub 整合 AI 工作流

第二步，整合其他工具，比如打开 [临时记事本](#)，你就可以用简洁的方式来规划 prompt，实现零干扰的 AI 沉浸空间。右边这段 Master 人设模版我已经注入了输入法，敲两下就能出来，你可以期待第 10 节课的分享。

高端的食材往往只要最朴素的烹饪方式

第三步，利用 ChatGPT 生成学习计划或项目计划。把计划粘贴到 Notion 中就会形成可追踪的里程碑，不再需要 To Do List 应用。这里可以作为团队成员的管理文档，拆解每天的工作。你还可以要求 ChatGPT 把生成的内容自动发送到 Notion 笔记中，提升团队沟通效率。这些目标都能够通过迭代达成。

用 ChatHub 制定可用计划

最后，如果团队里有图片相关的需求，智能绘图作为链接也能够流转到其他 AI 的生产线中。除了 Firefly，Midjourney 和 Stable Diffusion 的图片也可以这么用。当然，你要注意相关的版权细节，比如 \$10/ 月的 MJ 套餐生成的结果就不可商用。

智能绘图结果整合到 ChatGPT 文稿中

## 目光所及，强强联合，助你更高效

通过 ChatHub，你可以利用不同 AI 的优势去高效组合。怎么做呢？看看下面 AI 三剑客的截图，分别是 ChatGPT、Poe 上的 Claude、谷歌 Bard。

三剑客

最左边的 ChatGPT 我们比较熟悉了，智能时代的第一个冲浪板。大多时候可以优先在这个“人智协作舒适区”展开工作。不过，用 ChatGPT 就是最好的方案了吗？想要迁移 ChatGPT 的能力，继续往右看。

shikey.com转载分享

中间的 Poe 接入了 Claude、ChatGPT 和 GPT-4 等主流官方模型，不方便使用 OpenAI 服务的小伙伴不用再为访问问题犯愁了，打开就能用。假如有需要总结超长文本，那么它的 Claude-100K 模型可以胜任挑战。缺点是 Claude 的综合输出质量稍稍逊色。

最右，**免费的** Bard 胜在联网准确、生成速度一骑绝尘，默认生成 3 组结果参考，更有图像可视化的能力，大幅提升了可读性。缺点是大家比较关注的语言问题：目前只支持英文。

现在，ChatHub 里的 ChatGPT 和 Claude 都能轻松解决翻译问题，你可以同时让三剑客为你提供服务，不用切换多个 Tab，反复左右横跳。

## 综合对照，让 AI 有效内卷，为你所用

除了组合应用，还有一种提效场景是针对同一个问题做综合对照，交出最好的成果。

比如下面分别让 Claude、Bard 和 GPT-4 总结 K.K 新书的例子，初始 prompt 都是“总结 KK 的新书： *Excellent Advice for Living Wisdom I Wish Id Known Earlier*。”

首先验证是否可行。假如都顺利通过，你就定一个综合标准来评估哪一个结果是更好的，可以使用准确度、完整度、专业表达这些指标。

完成对照，你也可以用刚才说的“强强联合法”继续升华输出结果。Bard 有可视化的能力，要求结果里面先展示书籍封面就可以帮助快速确认是否瞎编。而 GPT-4 借助 AskYourPDF 插件就可以给到准确和详细的总结，再参考 Claude 的链接和 Bard 的封面，就能升级为丰富专业的输出。

总结 K.K 新书 Excellent Advice for Living Wisdom I Wish Id Known Earlier

## 小黑板

上手 ChatGPT 有三大里程碑：掌握独立设计 prompt、学会高质量和少冗余的提问、会迭代“答案”。这节课我们终于来到最后一个里程。

从“有错漏”和“不稳定”这 2 个普遍问题出发，我们打造了一个能够有效打通主流 AI 进行协作的 ChatHub。这个过程中，我们掌握了从源头到过程的迭代方法，还有针对成果的升级方法，打造了从生成方案代码到一键打通的沉浸式智能协作空间。**以前没有编程基础就完全不可能的方案，今天你在 ChatGPT 的帮助下就能做到。**

上手 ChatGPT 的三大里程碑

其实，迭代答案从初始 prompt 设计就开始了，我们灵活使用套路，直接出发。盘点好需求，结合 3W 原则，初始 prompt 就形成了。发出去之后，就要从源头开始观察生成结果。我推荐你使用 2 个“一键提效”技巧来纠偏，用“编辑”解决修错漏和补细节，用“重新生成”给你多组可对比的方案。如果方向没有偏离，就可以进入过程迭代，结合主动思考去引导 ChatGPT “打磨”，直到达成目标。

过程迭代我提供了实用的提示词技巧，核心作用是激活慢思维和提升推理质量。遇到阻碍和报错也不要紧，抓大放小、对照实验都是解决的方法，ChatGPT 会指引你走出困境。

迭代是为了更好的结果，但所有的结果都只是新的开始。不断突破的精神才是你给自己最大的惊喜。

**要点秒懂：**

## 课后练习

盘点一下自己的高频需求，有没有想过借助 ChatGPT 来重塑的解决方案呢？或者就是单纯实现更高效？尝试一下。可以是针对一个日常离不开的工具，通过这种整合方式和 ChatGPT 的内容生产全流程对接，可以是突然灵光一闪，打造一个取悦自己的智能骚操作。

期待在评论区看到你的思考或感受分享，也欢迎你将这节课分享给感兴趣的朋友们，戳此加入 [🔗 课程交流群](#)，我们下节课再会。

## 附：Arc 浏览器适用的 Python 代码

测试环境：MacOS

shikey.com转载分享

 复制代码

```
1 import subprocess
2
3 def open_tabs(urls):
4     for url in urls:
5         if 'notion.so' in url:
6             # 执行AppleScript脚本来打开新标签页并打开notion.so
7             script = 'tell application "Arc" to tell window 1 to make new tab wit
8             subprocess.run(['osascript', '-e', script], capture_output=True, text
9         else:
10            # 执行AppleScript脚本来打开新标签页并打开链接
11            script = 'tell application "Arc" to tell window 1 to make new tab wit
12            subprocess.run(['osascript', '-e', script], capture_output=True, text
13
14    print("🔔 任务已执行")
15
16 if __name__ == '__main__':
17     urls = ['https://ai.com', 'https://notion.so', 'https://bard.google.com', 'ht
18     open_tabs(urls)
```

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

## 精选留言 (3)

JER

2023-06-15 来自广东

这节课我们在群里面整了个ChatGPT抽奖，送出了两个Arc邀请码。分享一下prompt和会话。

原会话：<https://chat.openai.com/share/da2f3bfb-704d-4171-b5e2-99f7937f8a54>

你将扮演一个公平的抽奖器，从下面的ID中抽出2位幸运小伙伴：

1. ID
2. Y024
3. ReadU

4. 一阴一阳之谓道
5. Nothing
6. 追
7. zhjk
8. 我不是小朋友
9. 月涩风潇
10. 线

shikey.com转载分享

输出模版:

## 🎉 恭喜这两位朋友

#表格输出2个ID

获得Arc浏览器邀请码, 请小窗JER领取



👍 3



**石云升**

2023-06-29 来自广东

我收回我在发刊词里留言说的, 这门课很好学, 速来。

我要改成, 这门课学起来不容易, 但有好东西。同学们, 速来。

作者回复: 基础很容易, 后面灵活组合基础就需要实践吃透和思考了。这门课都是实践里萃取的精华和思考, 并不是简简单单的操作手册。



👍 1

**Toni**

2023-06-14 来自瑞士

ChatGPT生成的代码和对代码的解释: (Windows, Microsoft Edge)

如果你希望通过命令行 (CMD) 来打开 Microsoft Edge 浏览器并同时打开指定的标签页, 你可以使用以下的批处理脚本。

首先, 创建一个名为 open\_tabs.cmd 的文本文件, 将以下内容复制粘贴到文件中:

-----  
@echo off

```
set urls="https://firefly.adobe.com" "https://poe.com" "https://bard.google.com" "https://notion.so" "https://ai.com"
```

```
for %%i in (%urls%) do (
```

```
start microsoft-edge:%%i
)
echo 🎉 任务已执行
```

-----shikey.com转载分享

然后，保存文件并将其双击运行，或在 CMD 中执行该批处理脚本。

这个脚本会使用 start 命令打开 Microsoft Edge 浏览器，并通过使用 microsoft-edge: 协议打开指定的标签页。%urls% 变量包含了要打开的标签页的URL列表。循环遍历这些URL，并使用 start microsoft-edge:URL 命令来在 Microsoft Edge 中打开每个标签页。

执行完所有的命令后，会打印出 "🎉 任务已执行" 表示脚本成功完成了任务。

请注意，这种方法假定你的系统上已经安装了 Microsoft Edge 浏览器，并且它是默认浏览器。

希望能帮到你！如果你还有其他问题，请随时提问。

-----

使用本课的方法，ChatGPT生成的代码跑得通。

作者回复：只要清晰表达需求，ChatGPT不会让人失望。😄

共 3 条评论 >

