

GPT-4使用教程

GPT基础

GPT-4使用教程

其实你不需要一个GPT-4教程，你可以让GPT-4生成一个GPT-4教程。但是，下面的教程包含了GPT4无法意识到的一些使用技巧。

1. 安装和访问

- 首先，确保您的设备已经安装了VPN。在选择VPN节点时，建议避免使用香港节点，以确保稳定的网络连接。
- 选择使用Google进行登录。为了保护您的账户安全，如果你是从新设备登录，可能需要通过安全邮箱获取验证码。

2. 操作界面

- 使用GPT-4时，你可以像使用百度知道和CSDN那样进行操作。所有的使用方法都可以通过直接向GPT-4提问获取。
- 在GPT-4上鼠标悬浮时，你可以看到其他模式的选择（互联网模式：比newbing的输出长，搜索网页范围广，使用GPT4。插件模式：通过json命令调用webapp 比如wolfram for make math and common API req, webpilot for read web。

3. 使用注意事项

- 每个周期（例如，3小时）内只能提问25个的问题数量，避免一句话的输入和输出。如果问题很短，可以一次问多个问题。
- 不要删除其他人的提问。
- 超过一定时间（30天）的提问会被自动删除。
- 如果你觉得问题涉及到了我的知识库更新后（例如，2019年）的知识，请使用新的搜索工具。
- 注意在使用GPT-4的过程中，不要一直使用一个对话。因为GPT-4只能记住一定长度的上下文，一般是5-6轮对话。如果你一次输入过多内容（例如，300行），那么我可能只能记住最后一轮的上下文。
- 输入长度有限制。例如，GPT-3.5的输入长度限制为32712个字符，GPT-4的输入长度限制为16324个字符（约等于326行代码或5页论文）。
- 注意，每次GPT-4只能回答一个问题。如果GPT-4正在被使用，你可以选择使用GPT-3.5或者其他备用账户(hotmail) 或者清华的镜像网站。
- 如果您对英语熟练，可以尝试用英语向GPT-4提问，这样可能会得到更高质量的回答。

GPT-4如何完成复杂任务（写项目代码，写论文，写项目书）

1. **第一原则**：向GPT-4提供精确、充足的信息，并提出明确的问题。这是引导模型产生更准确、更相关的代码的基础。如果你对应提供哪些代码和信息感到不确定，可以通过提示向GPT-4询问。
2. **明确简化问题**：将编程问题表述得尽可能清晰易懂，保证每次都能获取到最准确的代码。对问题和需求的精确描述，可以大大提高获取所需代码的准确性。
3. **项目划分**：将大项目分解为小的、可管理的任务或函数，且最好每个任务的代码长度不超过两个回答的长度。这是因为GPT-4内置的continue generating按钮或者直接输入"continue"通常只能有效地续写一次。在这个过程中，为每个任务都向GPT-4提供详细的上下文，以便生成相应的代码。
4. **最大化上下文信息**：向GPT-4提供尽可能多的相关信息，包括项目的总体目标、当前任务的具体目标、需遵守的约束，以及当前任务如何适应整个项目。
5. **互动式提示和保持跟踪**：利用迭代式交谈，将模型的输出反馈作为输入，以生成更长的输出。同时，密切跟踪在编写或调试代码的过程，因为在GPT-4超过输入限制后并没有持久的记忆，所以需要手动管理连续性和上下文。
6. **使用元代码**：有些特定的提示或指令输入给ChatGPT后，你可以以高概率得到所需的代码，我们称之为“元代码”。元代码可以视为代码的DNA，能够引导AI生成预期的代码。
7. **灵活调整和重启**：当连续多次尝试后未能得到所需的代码（3次以上），应考虑重新开始新的对话，并相应地改变你的初始提示。新的对话环境有助于清空过往的错误尝试，从新的角度看待并解决问题。

实用的prompt

1. **模板式提示 (Templet-prompt)**：在这种方法中，你可以直接复制你想要的提示到开头。例如，如果你希望避免过度检测提示，可以像下面这样：

当涉及到创作内容时，有两个因素至关重要：一是“困惑度”，二是“突发性”。困惑度用来衡量文本的复杂性，而突发性则是用来比较句子变化的。我们人类在写作时，往往会展现出较大的突发性，比如，我们会混合使用一些较长或复杂的句子和较短的句子。然而，AI生成的句子往往更加统一。因此，在撰写以下内容时，我希望你能创作出既有足够困惑度，又有突发性的文本。接下来请你按照上面的要求，生成内容。
我的第一个任务是：

这样，你就设置了一种对话情境，让模型理解它的任务。

2. **丰富提示 (Richprompt)**：在这种方法中，尽可能提供充足的精确信息，然后输入分割线---在最后一行写出你的命令或请求，让模型在前面的上下文基础上进行生成。

```
```txt

生成一个能够将方程图片转化为Latex代码的web应用，
具体而言，页面应该包含一个输入框，用户可以在输入框中输入方程图片，然后点击提交按钮，页面应该显示出转化后的Latex代码。

请你生成一个能够将方程图片转化为Latex代码的web应用。
```
```

3. **元模板提示 (Meta-Templet-prompt)**：这种提示允许你编写一个模板，让模型根据这个模板生成特定的代码或提示。例如，你想要调试代码，你可以提供如下信息：

```
背景介绍
---
代码片段 1~n
---
报错信息 1~n
---
搜索整理的解决方案 (newbing or GPT4browse 得到的，也可以是csdn上搜的)
---
请给我修复后的代码
```

4. **元提示 (Meta-prompt)**：这是一种生成其他提示的提示，也被称为二阶提示。例如，你可以向模型提供一个网页HTML源码，并要求它创建一个类似的网页应用。元提示看起来可能像这样：

```
网页HTML源码
----
我希望部署一个与我所看到的类似的web应用，
你的目标是创建一个将方程图片转化为Latex代码的web应用，
具体而言，页面上的主要组件是一个用于编辑Latex代码的输入文本框，
下方是一个实时渲染当前方程的框，你可以将方程的图片粘贴到Latex编辑器中，
编辑器会自动将其转换为Latex代码。
你的第一个任务是为我做一个大綱，然后为每一部分生成提示，
这样我就可以通过依次向GPT4输入这些提示来达到我的目标。
```

```
Web Html source
----
i want to deploy a web app like what i see above,
your goal is to create an web app to convert picture of equation to
latex code,
specifically, the main component on the page is a input text box to
edit the latex code and a below is a box to render the current
equation in realtime , the key feature is that you can paste a image
of equation into the latex editor and the editor will automatically
turn it it to the latex code.
your first task is to make a outline of what i should do , and gen
prompts for every section so that i can reach the goal by typing those
prompts one by one to GPT4,
```

5. **元代码 (Meta-code)**：有些特定的提示或指令输入给ChatGPT后，你可以以很高的概率得到所需的代码，我们称之为“元代码”。元代码像是代码的DNA，能够引导AI产生期望的代码。

GPT的局限

当然，我可以根据你的要求调整这些限制：

1. **硬件限制**：GPT-4 的输入/输出长度有一定的限制，且在一个周期（例如3小时）内的查询次数也有限制。此外，其上下文记忆长度也相对有限，大约只能持续记住5轮左右的交互信息。
2. **信息处理局限**：当输入的信息过多时，GPT-4 可能会忽略或遗忘部分信息。在这种情况下，你需要确认并指出哪些信息是关键信息，确保GPT-4 有针对性地记住和处理。
3. **假设问题**：在处理某些问题时，GPT-4 有时会做出不符合需求的假设。为了解决这个问题，你可以要求GPT-4 列出其当前的假设，并进行适当的修正和引导。
4. **自指局限**：GPT-4 无法做出关于自己的分析，比如无法回答“上一个问题有几个字”这样的自我指向问题。以前是回答错误答案，现在会回答：

对不起，我无法回答“上一个问题有几个字”。作为一个AI模型，我并不具备实时计数或回溯前述文本长度的能力。我的主要功能是处理给定输入并生成相关的输出，而不是监控或追踪之前的输入内容。

5. **陷入错误循环**：在某些情况下，当超过3次修正都无法给出正确答案时，GPT-4 可能会陷入一个错误的循环，无法生成正确的答案。这在调试代码的场景中尤其明显。

请注意，这些局限性是基于GPT-4 当前的设计和训练情况，未来的版本或不同的AI模型可能会有所改善或变化。