

# DeepSeek 分享

DeepSeek 分享 by 余乐

Where the world  
builds software

Millions of developers and companies build, ship, and  
maintain their software on GitHub—the largest and  
most advanced platform in the world.

# DeepSeek 定位与价值

## 1. 模型背景

- 国产大模型代表，深度求索（DeepSeek Inc.）研发
- 技术定位：通用型LLM，强调多任务处理与高推理效率
- 应用领域：文本生成、代码生成、逻辑推理、知识问答等

## 2. 为什么要关注 DeepSeek?

- 差异化场景覆盖（如代码能力、长文本理解）
- 部署灵活性（支持API调用与私有化部署）
- 性价比优势（资源消耗与性能平衡）

# 私有化部署

## Ollama

<https://ollama.com/>

Ollama 是一个开源的大型语言模型（LLM）平台，旨在帮助用户在本地环境中轻松运行、管理和与各种预训练的语言模型交互。该平台支持多种自然语言处理任务，如文本生成、翻译、代码编写和问答等。

主要特点：

- 开源免费：Ollama 及其支持的模型完全开源，用户可以自由使用和修改。
- 简单易用：提供简洁的命令行界面和 API，用户无需复杂配置即可快速加载和使用各种预训练模型。
- 本地部署：允许在本地计算环境中运行模型，保障数据隐私，并降低对外部服务器的依赖。
- 多平台支持：兼容 macOS、Windows 和 Linux 操作系统，满足不同用户的需求。
- 丰富的模型库：支持多种热门开源模型，如 Llama 3.3、DeepSeek-R1、Phi-4、Mistral 和 Gemma 2 等，用户可以根据需要下载和切换模型。

## DeepSeek 不同版本

1.5b, 7b, 8b, 14b, 32b, 70b, 671b 指的是 Billion(十亿), 代表模型的参数

<https://ollama.com/library/deepseek-r1>

## 一、消费级设备配置

适用于个人开发者或中小型项目，主要覆盖 **1.5B** 至 **32B** 参数模型：

模型参数	显存需求	GPU 推荐	CPU 与内存	适用场景
<b>1.5B</b>	2-4 GB	NVIDIA GTX 1660	8 GB 内存 + i5/Ryzen5	轻量级文本生成、简单问答
<b>7B</b>	8-12 GB	RTX 3060/4060	16 GB 内存 + i7/Ryzen7	代码生成、多轮对话
<b>14B</b>	16-24 GB	RTX 3090/4090	32 GB 内存 + i9/Ryzen9	复杂推理、长文本处理
<b>32B</b>	24-32 GB	RTX 4090 (单卡)	64 GB 内存 + Xeon/EPYC	深度思考、专业领域问答

关键说明：

- 显存优化：通过 **量化技术**（如 4-bit/8-bit）可降低显存占用，例如 32B 模型使用 4-bit 量化后显存需求可降至 16-20 GB。
- 多卡支持：32B 及以上模型可通过多张 RTX 4090 (24GB 显存) 并行运行，但需配置 PCIe 4.0 高带宽通道。

## 二、企业级设备配置

适用于大规模推理或专业场景（如金融、医疗），覆盖 **70B** 至 **671B** 参数模型：

模型参数	显存需求	GPU 推荐	CPU 与内存	适用场景
<b>70B</b>	64 GB+ (单卡)	NVIDIA A100/H100	128 GB 内存 + 多路 Xeon	大规模数据分析、复杂决策
<b>671B</b>	多卡 80 GB+/卡	8-16 张 A100/H100	256 GB+ 内存 + 服务器级CPU	超大规模分布式推理

关键说明：

- 硬件协作：671B 满血版需通过 **多 GPU 集群** 实现模型分片加载，推荐使用 NVIDIA 的 NVLink 或 InfiniBand 技术提升通信效率。
- 存储要求：671B 模型文件体积约 642 GB (FP8 精度)，需搭配高速 SSD (如 PCIe 5.0) 避免 I/O 瓶颈。

## 运行

```
ollama run deepseek-r1:14b
```

通过上面命令直接运行 DeepSeek-R1:14 模型，不存在此模型的时候会直接拉去模型文件

这个时候就可以在 terminal 终端开始问答

但是在命令行中进行交互始终不方便，所以我们需要一个 GUI 的客户端

# Page Assist

Page Assist 浏览器插件

<https://github.com/n4ze3m/page-assist>

# Open WebUI

<https://github.com/open-webui/open-webui>

## 1. 通过 docker 运行

```
docker pull ghcr.io/open-webui/open-webui:main
docker run -d -p 3000:8080 -v open-webui:/app/backend/data --name open-webui ghcr.io/open-webui/open-webui:main
```

<http://localhost:3000>

# 基础功能演示

## 文本生成

- 创意写作

任务：写一篇关于“未来城市”的科幻短文。

- 营销文案

任务：为一家咖啡店写一段吸引顾客的广告文案。

## 翻译

找一段内容, 将英语翻译成中文

<https://github.com/deepseek-ai/DeepSeek-R1?tab=readme-ov-file#2-model-summary>

## 代码生成

- Python/JS 代码生成

任务: 写一个Python函数，计算斐波那契数列的第n项。 将上面的 Python 代码改成 JavaScript 代码

- SQL 语句生成

任务: SQL查询: “查找订单金额大于1000的客户姓名”。 订单表 `orders` 用户表 `customers`

## 知识问答：法律与医疗领域

- 问题：在中国，劳动合同法规定试用期最长是多久？
- 问题：糖尿病患者应该如何控制饮食？

## 多轮对话：上下文连贯性测试

1. 我想去北京旅行，有什么推荐的地方吗？
2. 你可以帮我制定一个旅游路线计划吗，我大约在北京呆 2 天

## 附件上传

上传附件 `financial\_report.xlsx`

提问: 根据附件 `financial\_report.xlsx` , 张三的营业额是多少?

上传附件 `compony\_policy.docx`

- 提问: 根据文档中的描述, 请假流程是? 带薪休假有多少天?
- 提问: 根据附件中 `compony\_policy.docx` , 生成一句话总结公司政策。

## Prompt

- Prompt工程优化：结构化指令设计（角色设定、分步思考链）
- 在 Prompt 工程中，结构化指令设计可以帮助模型更好地理解任务，并生成更精准的回复。
- 以下是两个具体的案例和优化示例：

# 案例1：角色设定优化

场景：你需要让模型扮演一位经验丰富的产品经理，帮助你设计一款新的移动应用。

普通Prompt：“帮我设计一款移动应用。”

优化后的Prompt：“你是一位拥有10年经验的产品经理，专注于用户体验和市场需求分析。请根据以下步骤设计一款移动应用：

目标用户：明确应用的目标用户群体及其核心需求。

核心功能：列出应用的3个核心功能，并解释其价值。

竞品分析：分析市场上类似应用的优缺点。

用户体验：描述应用的主要界面设计和用户交互流程。

商业模式：建议一种可行的商业模式（如订阅制、广告等）。请分步回答，确保逻辑清晰。”

优化点：

通过角色设定（经验丰富的产品经理），限定了模型的回答风格和专业性。

分步思考链（目标用户→核心功能→竞品分析→用户体验→商业模式）引导模型结构化输出，避免遗漏关键点。

## 案例2：分步思考链优化

场景：你需要让模型帮助你解决一个复杂的数学问题。

普通Prompt：“解方程： $2x + 5 = 15$ 。”

优化后的Prompt：“你是一位数学老师，请按照以下步骤解方程：

理解问题：明确方程的目标是求解变量x的值。

简化方程：将方程两边减去5，得到 $2x = 10$ 。

求解变量：将方程两边除以2，得到 $x = 5$ 。

验证结果：将 $x = 5$ 代入原方程，验证等式是否成立。请分步解释每一步的操作和原理。”

优化点：

通过分步思考链（理解问题→简化方程→求解变量→验证结果），确保模型不仅给出答案，还解释每一步的逻辑。

角色设定（数学老师）让模型的回答更具教育性和权威性。

# prompts 资源

<https://github.com/PlexPt/awesome-chatgpt-prompts-zh>

# DeepSeek API 调用

<https://api-docs.deepseek.com/zh-cn/>

```
import OpenAI from "openai"
import 'dotenv/config'

const openai = new OpenAI({
  baseURL: 'https://api.deepseek.com',
  apiKey: process.env.API_KEY,
})

async function main(message) {
  const completion = await openai.chat.completions.create({
    messages: [
      { role: "system", content: "You are a helpful assistant." },
      { role: 'user', content: message },
    ],
    model: "deepseek-chat",
  })

  console.log(completion.choices[0].message.content)
}

main('DeepSeek 是什么？做一个简短的介绍，他有那些优势和那些不足？')
```

# Ollama 接口调用

提问: 我使用 Ollama 本地跑了一个 deepseek 模型, 我如何通过 node.js 调用

```
// https://github.com/ollama/ollama/blob/main/docs/api.md#generate-a-completion

// curl http://localhost:11434/api/generate -d '{
//   "model": "llama3.2",
//   "prompt": "Why is the sky blue?"
// }'

const question = '简单介绍下你是那个模型'
const url = 'http://localhost:11434/api/generate'
const data = {
  model: 'deepseek-r1',
  prompt: question,
  stream: false,
}

fetch(url, {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(data)
})
  .then(response => response.json())
```

# 问答环节