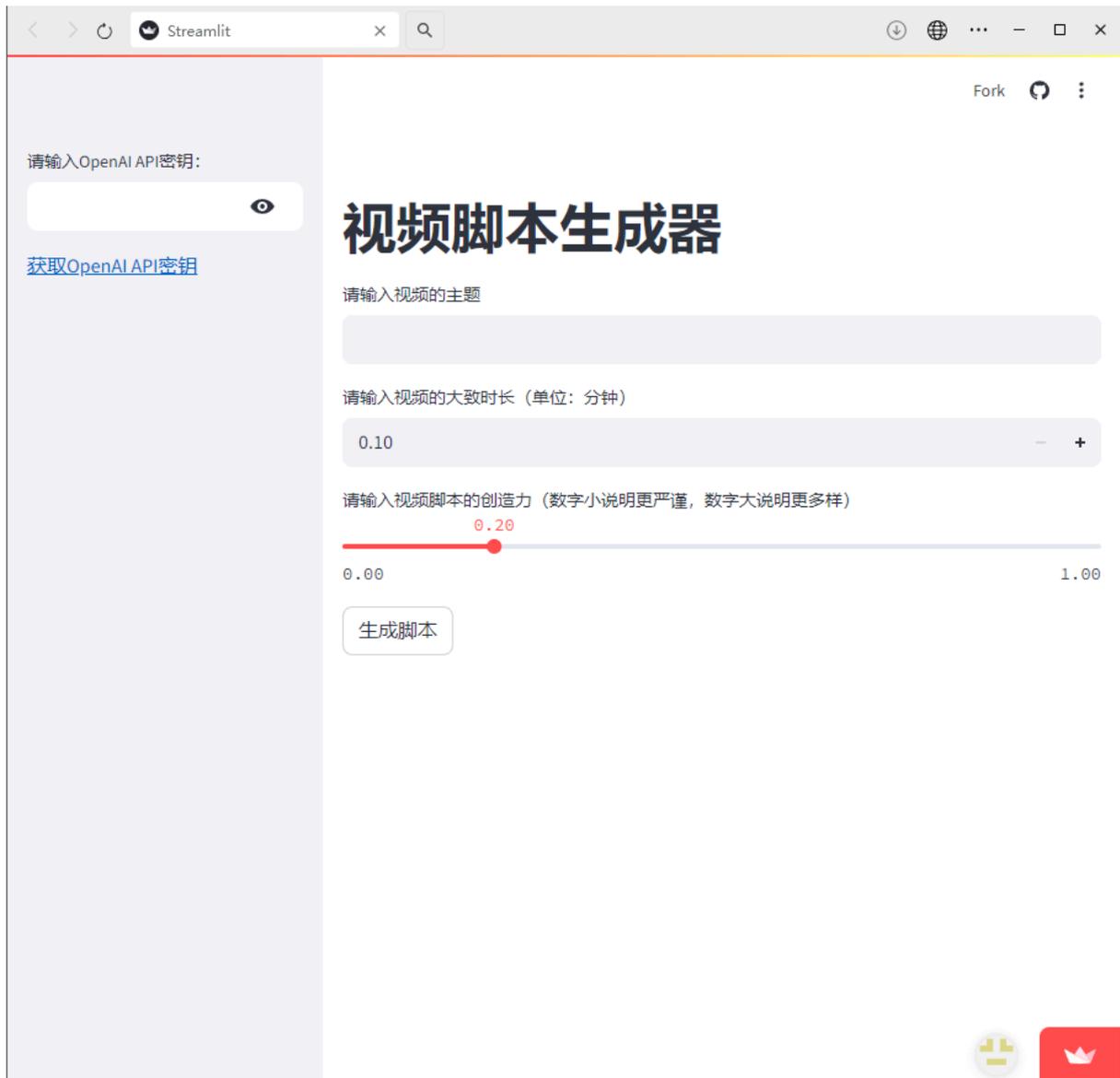


视频脚本生成器网址： <https://generatescript-openai.streamlit.app/>

🕒 创建时间	@2025年7月17日 17:35
☰ 标签	

- ✨ 网页需要科学上网才可见 许久不打开需要手动刷新
- ✨ 本视频脚本生成器调用OpenAI API
- ✨ 下方为前后效果图



请输入OpenAI API密钥:

..... [获取OpenAI API密钥](#)

视频脚本生成器

请输入视频的主题

Deepseek对中国的影响力

请输入视频的大致时长 (单位: 分钟)

0.10

- +

请输入视频脚本的创造力 (数字小说明更严谨, 数字大说明更多样)

0.20

0.00

1.00

生成脚本

视频脚本已生成

标题:

《Deepseek: 揭开中国影响力的新篇章》

视频脚本:

【开头】 (镜头快速切换, 背景音乐轻快)

大家好, 欢迎来到我们的频道! 今天我们要聊的是一个超级酷的话题——《Deepseek: 揭开中国影响力的新篇章》。你有没有想过, 为什么中国的影响力在全球越来越大? 今天就让我们一起揭开这个神秘的面纱!

【中间】 (镜头切换到图表和数据)

首先, Deepseek是一个新兴的科技平台, 专注于数据分析和信息挖掘。它通过强大的算法, 帮助我们更好地理解中国在全球的经济、文化和科技影响力。比如, 最近的数据显示, 中国在人工智能领域的投资已经超过了其他国家的总和, 这可不是开玩笑的哦!

(镜头切换到年轻人讨论的场景)

而且, Deepseek还帮助年轻人找到更多的机会, 无论是创业、就业还是学习新技能。想象一下, 利用这些数据, 你可以在职场上走得更远, 甚至成为行业的领军人物!

【结尾】 (镜头回到博主, 神秘微笑)

所以, 想要了解更多关于中国影响力的秘密吗? 别忘了关注我们, 下一期我们将揭晓一个更大的惊喜——如何利用这些数据改变你的人生! 快来一起探索吧!

(背景音乐渐弱, 画面渐出)



```
from langchain.prompts import ChatPromptTemplate
from langchain_openai import ChatOpenAI
from langchain_community.utilities import WikipediaAPIWrapp
```

```

er
from openai import api_key
# import os

#设置一个函数 这样之后就可以通过调用它来得到脚本的内容
def generate_script(subject,video_length,creativity,api_key):
#设置标题提示模版和脚本提示模版
    title_Template=ChatPromptTemplate.from_messages(
        [
            ("human","请为{subject}这个主题的视频想一个吸引人的标题")
        ]
    )
    Script_Template=ChatPromptTemplate.from_messages(
        [
            ("human",
            """你是一位短视频频道的博主。根据以下标题和相关信息，为短视频频道写一个视频脚本。
            视频标题：{title}，视频时长：{duration}分钟，生成的脚本的长度尽量遵循视频时长的要求。
            要求开头抓住眼球，中间提供干货内容，结尾有惊喜，脚本格式也请按照【开头、中间，结尾】分隔。
            整体内容的表达方式要尽量轻松有趣，吸引年轻人。
            脚本内容可以结合以下维基百科搜索出的信息，但仅作为参考，只结合相关的即可，对不相关的进行忽略：
            ```{wikipedia_search}```""")
]
)
#调用api
 model=ChatOpenAI(model="gpt-4o-mini",openai_api_key=api_key,temperature=creativity)
#把提示词模版和ai连接起来
 title_chain = title_Template | model
 Script_Chain=Script_Template | model

#引入参数
 title=title_chain.invoke({"subject":subject}).content

```

```

search=WikipediaAPIWrapper()
search_result=search.run(subject)

script=Script_Chain.invoke({"title":title,"duration":video_length,"wikipedia_search":search_result}).content
#返回得到的内容
return title,search_result,script

print(generate_script("sora模型", 1, 0.7, os.getenv("OPEN AI_API_KEY")))

```

```

#从之前的后端代码文件中引入generate_script
import streamlit as st
from utils import generate_script

st.title("视频脚本生成器")
#侧边栏弄一个输入API密钥的
with st.sidebar:
 openai_api_key= st.text_input("请输入OpenAI API密钥: ",type="password")
 st.markdown("[获取OpenAI API密钥](https://openai.com/api/)")
#设置一些输入参数的
subject= st.text_input("请输入视频的主题")
video_length= st.number_input("请输入视频的大致时长（单位：分钟）", min_value=0.1,step=0.1)
creativity=st.slider("请输入视频脚本的创造力（数字小说明更严谨，数字大说明更多样）",min_value=0.0,value=0.2,max_value=1.0,step=0.1)
#设置提交按钮
submit= st.button("生成脚本")
#如果上面没有达成条件 要求重新输入 并且停止运行程序
if submit and not openai_api_key:
 st.info("请输入你的OpenAI API密钥")
 st.stop()
if submit and not video_length:
 st.info("请输入视频的大致时长")
 st.stop()

```

```
if submit and not subject:
 st.info("请输入视频的主题")
 st.stop()
#上面满足条件则.....一个旋转按钮表示思考，将之前generate_script返回的title,search_result,script赋值给title,search_result,script，注意要一一对齐，要么可能出现标题那里出现维基百科信息的情况
if submit:
 with st.spinner("AI 思考中.....请稍等"):
 title,search_result,script = generate_script(subject,video_length,creativity,openai_api_key)
 st.success("视频脚本已生成")
 st.subheader("标题：")
 st.write(title)
 st.subheader("视频脚本：")
 st.write(script)
 with st.expander("维基百科搜索结果"):
 st.info(search_result)
```