

# 追踪我们的家庭财务

最近，我创建了一个 Markdown 文档来记录我们家庭的财务交易。

这套房子属于我和妻子。我的父母给了我们一些钱，我们还向我的姐姐和舅舅借了钱。虽然舅舅把钱转给了我，但后来我父亲偿还了那笔款项。

我们支付了房屋总价的 50% 作为首付，另一半则从中国农业银行贷款。贷款合同期限为 20 年，目前的利率是 3.65%。

在我失业期间，我的妻子和父亲为我提供了资金，以支付每月的房贷。因此，涉及到了许多交易。

我主要使用招商银行作为我的主要银行。招商银行允许根据交易是收入还是支出以及最小金额进行筛选。它还支持通过关键词筛选，这非常有帮助。

另一个有用的方面是人工智能的普及。它也能协助完成这项任务。利用 AI 驱动的 OCR 技术，特别是 Grok，我能够从与广州电力局的交易记录中提取文本。

由于后面的表格是基于前面的数字，最好先检查数字，确保一切正确后再继续。

下面的代码帮助从 Markdown 生成 PDF。它有一些特殊设置，以支持 PDF 中的中文字符渲染。

```
import os
import subprocess

# 配置
CJK_FONT = "Heiti SC"
GEOMETRY = "margin=1in"
input_markdown_path = "mortgage.md" # 替换为你的输入 Markdown 文件
output_pdf_path = "mortgage.pdf" # 替换为你期望的输出 PDF 文件

# 检查输入文件是否存在
if not os.path.exists(input_markdown_path):
    raise Exception(f" 输入文件不存在: {input_markdown_path}")

# 构建 Pandoc 命令
command = [
    'pandoc',
    input_markdown_path,
    '-o', output_pdf_path,
    '-f', 'markdown',
```

```

    '--pdf-engine', 'xelatex',
    '-V', f'CJKmainfont={CJK_FONT}',
    '-V', f'CJKsansfont={CJK_FONT}',
    '-V', f'CJKmonofont={CJK_FONT}',
    '-V', f'geometry:{GEOMETRY}',
    '-V', 'classoption=16pt',
    '-V', 'CJKOptions=Scale=1.1',
    '-V', 'linestretch=1.5'
]

# 运行 Pandoc 命令
try:
    subprocess.run(command, check=True)
    print(f"PDF 成功生成: {output_pdf_path}")
except subprocess.CalledProcessError as e:
    print(f"生成 PDF 时出错: {e}")

```