

# 追蹤我們的家庭財務

最近，我創建了一個 markdown 文件來記錄我們房子的財務交易。

這房子屬於我和我妻子。我的父母給了我們錢，我們還向我姐姐和我的舅舅借了錢。雖然我的舅舅把錢轉給了我，但後來我父親還了那筆錢。

我們支付了房子總價的 50% 作為首付，另外一半是從中國農業銀行借的。合同期限為 20 年，目前的利率是 3.65%。

當我失業時，我的妻子和父親提供了資金來支付每月的房貸。因此，涉及了許多交易。

我使用招商銀行作為我的主要銀行。招商銀行允許按交易是收入還是支出以及最小金額來過濾交易。它還支持按關鍵字過濾，這非常有幫助。

另一個有幫助的方面是 AI 的普及。它也可以協助完成這項任務。使用 AI 驅動的 OCR，特別是 Grok，我能夠從與廣州電力局的交易記錄中提取文本。

由於後面的表格是基於前面的數字，最好先檢查數字，確保一切正確，然後再繼續。

下面的代碼幫助從 markdown 生成 PDF。它有一些特殊設置來支持 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"    # 替換為你的輸入 Markdown 文件
output_pdf_path = "mortgage.pdf"      # 替換為你想要的輸出 PDF 文件

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

# 構建 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'
]
```

```
# 運行 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 時出錯: {e}")
```