

# 深度搜尋實驗

## 主要要點

- 似乎在腳本中添加 `create-original` 命令會直接在 `_posts/en` 目錄中創建 Markdown 文件，格式與草稿文件相同，包括文件名中的日期。
- 研究表明，新命令將使用當前日期（2025 年 2 月 27 日），並遵循命名約定 `{date_str}-{name}-en.md`，與草稿創建過程匹配。
- 證據顯示，前置資料將與草稿相同，包括字段如 `audio: false` `lang: en` `layout: post` `title: {name}` 和 `translated: false` `。`

## 直接回答

**概述** 此回應在您的 Python 腳本中添加了一個 `create-original` 命令，允許您直接在 `_posts/en` 目錄中創建 Markdown 文件，繞過草稿過程。文件將遵循與草稿文件相同的格式，包括文件名中的當前日期，並保持相同的前置資料結構。

**運作方式** 新的 `create-original` 命令將：  
- 使用今天的日期，2025 年 2 月 27 日，格式為 `YYYY-MM-DD`（例如 `2025-02-27`）。  
- 在 `_posts/en` 目錄中創建文件，文件名如 `2025-02-27-{name}-en.md`，其中 `{name}` 是提供的名稱（例如 `test-post` 變為 `2025-02-27-test-post-en.md`）。  
- 包含與草稿文件相同的前置資料，例如：  
`--- audio: false lang: en layout: post title: {name} translated: false ---`  
- 確保 `_posts/en` 目錄存在，必要時創建它。

**意外細節** 有趣的是，儘管草稿文件和新原始文件共享相同的基於日期的命名約定，但現有的 `delete_md` 函數尋找沒有日期的文件（例如 `{name}-en.md`），這可能會導致文件管理中的不一致。您可能需要更新刪除邏輯以處理帶日期的文件名，以實現完全兼容。

---

## 調查筆記：添加 `create-original` 命令的詳細分析

本節提供了對在提供的 Python 腳本中實施 `create-original` 命令的全面分析，擴展了直接回答，提供了對腳本結構、實施背後的原因以及潛在影響的詳細見解。該分析基於腳本的現有功能和用戶請求添加一個新命令，該命令在“原始目錄”中直接創建文件，格式與草稿文件相同。

**背景與上下文** 位於“scripts”目錄中的腳本，名為“file.py”，處理 Markdown 文件的創建和刪除，這些文件似乎是用於多語言博客或內容管理系統，可能使用靜態網站生成器如 Jekyll。它目前支持三個命令：  
- `create`：在 `_drafts` 目錄中創建一個包含當前日期的草稿 Markdown 文件，例如 `2025-02-27-{name}-en.md`。  
- `create-note`：在 `notes` 目錄中創建一個包含日期的筆記文件。  
- `delete`：從 `_posts` 目錄及相關資產目錄中刪除 Markdown 文件、PDF 文件和音頻文件，尋找名為 `{name}-{lang}.md` 且沒有日期的文件。

用戶請求添加一個 `create-original` 命令，該命令在“原始目錄”中直接創建文件，保持與默認草稿創建 (`create` 命令) 相同的格式。根據腳本的結構和 `delete_md` 函數的行為，“原始目錄”被解釋為 `_posts/en`，即英文文章的目錄。

**實施細節** 為滿足請求，設計了一個新函數 `create_original`，模仿 `create_md` 函數，但針對 `_posts/en` 目錄。實施細節如下：

- **日期處理**：函數使用 `datetime.date.today()` 獲取當前日期，2025 年 2 月 27 日，04:00 AM PST，結果為 `2025-02-27`。這個日期格式為 `YYYY-MM-DD`，以與草稿文件名保持一致。
- **目錄和文件路徑**：函數檢查 `_posts/en` 目錄是否存在，必要時使用 `os.makedirs` 創建它。然後在 `os.path.join('_posts', 'en', f"{date_str}-{name}-en.md")` 創建文件，確保文件名包含日期，例如 `2025-02-27-test-post-en.md` 為名稱 `test-post`。
- **前置資料**：前置資料與 `create_md` 相同，定義如下：

```
---
audio: false
lang: en
layout: post
title: {name}
translated: false
---
```

這確保了與草稿文件的一致性，保留了如 `audio: false` 表示沒有音頻附件、`lang: en` 表示英文和 `title: {name}` 表示文章標題的字段。

- **文件創建**：使用 `open(file_path, 'w', encoding='utf-8')` 以 UTF-8 編碼寫入文件，確保廣泛兼容性，並打印確認消息，例如 `Created original file: _posts/en/2025-02-27-test-post-en.md`。

腳本的主要部分已更新以包含 `create-original` 作為有效操作，修改了使用消息為：

```
Usage: python scripts/file.py <create|create-note|create-original|delete> <name>
```

並添加了一個條件來在操作為 `create-original` 時調用 `create_original(name)`。

**與現有函數的比較** 為突出差異和相似之處，請考慮以下表格，比較 `create_md`、`create_note` 和新的 `create_original`：

函數名	目錄	文件名格式	前置資料字段	註釋
<code>create_md</code>	<code>_drafts</code>	<code>{date_str}-{name}-en.md</code>	<code>audio, lang, layout, title, translated</code>	創建英文文章的草稿文件

函數名	目錄	文件名格式	前置資料字段	註釋
create_note	notes	{date_str}-{name}-en.md	title, lang, layout, audio, translated	創建筆記文件，相似的前置資料
create_original_posts/en	{date_str}-{name}-en.md	audio, lang, layout, title, translated		新命令，與草稿相同格式，在文章中

此表格說明了 create\_original 在文件名格式和前置資料方面與 create\_md 一致，但針對 \_posts/en 目錄，繞過草稿階段。

**潛在影響和考量** 儘管實施滿足了用戶的請求，但有顯著的影響，特別是與現有的 delete\_md 函數：- **文件名不一致**：delete\_md 函數在 \_posts/lang 中尋找名為 {name}-{lang}.md 的文件，例如 \_posts/en/test-post-en.md，沒有日期。然而，create\_original 創建的文件包含日期，例如 \_posts/en/2025-02-27-test-post-en.md。這種不一致意味著 delete\_md 可能無法找到 create\_original 創建的文件，除非修改以處理帶日期的文件名，可能使用 glob.glob 並使用模式如 \*{-en,-zh,...}.md 來考慮日期。- **網站結構**：腳本暗示了一個多語言設置，\_posts 中的子目錄用於每種語言（en、zh 等），delete\_md 模式中缺少日期意味著 \_posts 中的文章可能不依賴文件名中的日期進行排序，可能使用前置資料或其他元數據。這與 Jekyll 基礎網站不同，文件名中的日期通常決定文章日期，但與腳本的當前行為一致。- **語言範圍**：實施專注於英文（lang: en），因為 create\_md 和用戶的請求暗示。如果用戶需要 create-original 針對其他語言，函數將需要擴展，類似於 delete\_md 以處理多種語言。

**理由和決策** 對“與默認創建草稿相同格式”的解釋至關重要。最初，對於這是指文件名格式（包括日期）還是前置資料存在不確定性。根據用戶對“格式”的強調和 create\_md 中的示例（包括文件名中的日期），決定 create\_original 使用 {date\_str}-{name}-en.md，確保一致性。然而，這個決定突出了一個潛在的與 delete\_md 的衝突，建議未來可能需要調整文件管理。

“原始目錄”根據腳本結構，特別是 delete\_md 在 \_posts/lang 的操作，推斷為 \_posts/en。這種假設與典型的博客設置一致，文章位於 \_posts，並且 /en 子目錄適合多語言上下文。

**結論** 添加 create-original 命令增強了腳本的功能，允許在 \_posts/en 中直接創建帶日期的 Markdown 文件，與草稿相同的前置資料。2025 年 2 月 27 日，這意味著將創建如 2025-02-27-{name}-en.md 的文件，保持與草稿格式的一致性。然而，用戶應該注意與 delete\_md 函數的文件名不匹配，這可能需要更新以實現完全兼容。

## 關鍵引用

- Python os 模組文檔
- Python datetime 模組文檔
- Jekyll 文章文檔