

深度搜索实验

关键点

- 似乎添加一个 `create-original` 命令到脚本中会直接在 `_posts/en` 目录中创建 Markdown 文件，格式与草稿文件相同，包括文件名中的日期。
- 研究表明，新命令将使用当前日期（2025 年 2 月 27 日），并遵循 `{date_str}-{name}-en.md` 的命名约定，与草稿创建过程匹配。
- 证据表明，前置信息将与草稿相同，字段如 `audio: false`、`lang: en`、`layout: post`、`title: {name}` 和 `translated: false`。

直接回答

概述 此响应向您的 Python 脚本添加了一个 `create-original` 命令，允许您直接在 `_posts/en` 目录中创建 Markdown 文件，绕过草稿过程。文件将遵循与草稿文件相同的格式，包括文件名中的当前日期，并保持相同的前置信息结构。

工作原理 新的 `create-original` 命令将：
- 使用今天的日期，2025 年 2 月 27 日，格式为 `YYYY-MM-DD`（例如，`2025-02-27`）。
- 在 `_posts/en` 目录中创建文件，文件名如 `2025-02-27-{name}-en.md`，其中 `{name}` 是提供的名称（例如，`test-post` 变为 `2025-02-27-test-post-en.md`）。
- 包括与草稿文件相同的前置信息，例如：
`---` `audio: false` `lang: en` `layout: post` `title: {name}` `translated: false` `----` 确保 `_posts/en` 目录存在，必要时创建它。

意外细节 有趣的是，尽管草稿文件和新的原始文件共享相同的基于日期的命名约定，现有的 `delete_md` 函数却寻找没有日期的文件（例如，`{name}-en.md`），这可能会导致文件管理中的不一致。您可能需要更新删除逻辑以处理带日期的文件名，以实现完全兼容。

调查笔记：添加 `create-original` 命令的详细分析

本节提供了对在提供的 Python 脚本中实现 `create-original` 命令的全面分析，扩展了直接回答，详细介绍了脚本的结构、实现背后的原因以及潜在的影响。分析基于脚本的现有功能和用户请求添加一个新命令，该命令在“原始目录”中直接创建与草稿文件格式相同的文件。

背景与上下文 脚本位于“scripts”目录中，名为“file.py”，处理创建和删除 Markdown 文件，似乎是为多语言博客或内容管理系统，可能使用静态站点生成器如 Jekyll。它目前支持三个命令：
- `create`：在 `_drafts` 目录中创建一个包含当前日期的草稿 Markdown 文件，例如 `2025-02-27-{name}-en.md`。
- `create-note`：在 `notes` 目录中创建一个包含日期的笔记文件。
- `delete`：从 `_posts` 目录及其相关资产目录中删除 Markdown 文件、PDF 文件和音频文件，查找没有日期的文件，例如 `{name}-{lang}.md`。

用户请求添加一个 `create-original` 命令，该命令在“原始目录”中直接创建文件，保持与默认草稿创建（`create` 命令）相同的格式。根据脚本的结构和 `delete_md` 函数的行为，“原始目录”被解释为 `_posts/en`，即英文帖子的目录。

实现细节 为了满足请求，设计了一个新函数 `create_original`，模仿 `create_md` 函数但针对 `_posts/en` 目录。实现细节如下：

- **日期处理：**函数使用 `datetime.date.today()` 获取当前日期，2025 年 2 月 27 日，格式为 YYYY-MM-DD，以与草稿文件名保持一致。
- **目录和文件路径：**函数检查 `_posts/en` 目录是否存在，必要时使用 `os.makedirs` 创建它。然后在 `os.path.join('_posts', 'en', f'{date_str}-{name}-en.md')` 创建文件，确保文件名包含日期，例如 `2025-02-27-test-post-en.md`。
- **前置信息：**前置信息与 `create_md` 中的相同，定义为：

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

这确保了与草稿文件的一致性，保留了字段如 `audio: false`（无音频附件）、`lang: en`（英文）和 `title: {name}`（帖子标题）。

- **文件创建：**使用 `open(file_path, 'w', encoding='utf-8')` 写入文件，确保 UTF-8 编码以实现广泛兼容性，并打印确认消息，例如 `Created original file: _posts/en/2025-02-27-test-post-en.md`。

脚本的主要部分被更新以包含 `create-original` 作为有效操作，修改了使用消息为：

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

并添加了一个条件以在操作为 `create-original` 时调用 `create_original(name)`。

与现有函数的比较 为了突出差异和相似之处，考虑以下表格比较 `create_md`、`create_note` 和新的 `create_original`：

函数	目录	文件名格式	前置信息字段	备注
<code>create_md</code>	<code>_drafts</code>	<code>{date_str}-{name}-en.md</code>	<code>audio, lang, layout, title, translated</code>	创建英文帖子的草稿文件

函数	目录	文件名格式	前置信息字段	备注
create_note	notes	{date_str}-{name}-en.md	title, lang, layout, audio, translated	创建笔记文件，类似前置信息
create_origin	_posts/en	{date_str}-{name}-en.md	audio, lang, layout, title, translated	新命令，与草稿相同格式，在帖子中

此表格说明 `create_original` 在文件名格式和前置信息上与 `create_md` 一致，但针对 `_posts/en` 目录，绕过草稿阶段。

潜在影响和考虑 虽然实现满足了用户的请求，但有一些显著的影响，特别是与现有的 `delete_md` 函数相关：
- 文件名不一致： `delete_md` 函数在 `_posts/lang` 中查找文件名为 `{name}-{lang}.md` 的文件，例如 `_posts/en/test-post-en.md`，没有日期。然而，`create_original` 创建带日期的文件，例如 `_posts/en/2025-02-27-test-post-en.md`。这种不一致意味着 `delete_md` 可能无法找到 `create_original` 创建的文件，除非修改以处理带日期的文件名，可能使用 `glob.glob` 与模式如 `*{-en,-zh,...}.md` 来处理日期。
- 站点结构： 脚本暗示一个多语言设置，`_posts` 中的子目录为每种语言（`en`、`zh` 等），`delete_md` 中没有日期的模式表明 `_posts` 中的帖子可能不依赖文件名中的日期进行排序，可能使用前置信息或其他元数据。这对于基于 Jekyll 的站点来说不寻常，因为文件名中的日期通常决定帖子日期，但这与脚本的当前行为一致。
- 语言范围： 实现专注于英文（`lang: en`），因为 `create_md` 和用户的请求暗示。如果用户需要 `create-original` 适用于其他语言，函数将需要扩展，类似于 `delete_md` 处理多种语言的方式。

理由和决策 对“与默认创建草稿相同的格式”的解释至关重要。最初，不确定这是指文件名格式（包括日期）还是前置信息。根据用户对“格式”的强调和 `create_md` 中的示例（包括文件名中的日期），决定 `create_original` 应使用 `{date_str}-{name}-en.md`，确保一致性。然而，这种决定突出了与 `delete_md` 的潜在冲突，建议未来可能需要对文件管理进行调整。

“原始目录”根据脚本的结构，特别是 `delete_md` 在 `_posts/lang` 中的操作，推断为 `_posts/en`。这种假设符合典型的博客设置，帖子位于 `_posts`，`/en` 子目录适合多语言上下文。

结论 添加 `create-original` 命令增强了脚本的功能，允许在 `_posts/en` 中直接创建带日期的 Markdown 文件，与草稿相同的前置信息。2025 年 2 月 27 日，这意味着将创建类似 `2025-02-27-{name}-en.md` 的文件，保持与草稿格式的一致性。然而，用户应注意与 `delete_md` 函数的潜在文件名不匹配，这可能需要更新以实现完全兼容。

关键引用

- Python os 模块文档
- Python datetime 模块文档
- Jekyll 文章文档