

深層検索実験

重要ポイント

- `create-original` コマンドをスクリプトに追加することで、`_posts/en` ディレクトリに直接 Markdown ファイルを作成し、ドラフトファイルと同じ形式で日付を含むファイル名を持つ可能性が高いです。
- 研究によると、新しいコマンドは現在の日付（2025年2月27日）を使用し、ファイル名の命名規則は `{date_str}-{name}-en.md` となり、ドラフト作成プロセスに合致します。
- 証拠は、フロントマターがドラフトと同じであることを示唆しており、フィールドは `audio: false`、`lang: en`、`layout: post`、`title: {name}`、`translated: false` です。

直接の回答

概要 この回答は、Python スクリプトに `create-original` コマンドを追加し、`_posts/en` ディレクトリに直接 Markdown ファイルを作成することで、ドラフトプロセスをバイパスします。ファイルはドラフトファイルと同じ形式で、ファイル名に現在の日付を含み、同じフロントマター構造を保持します。

仕組み 新しい `create-original` コマンドは以下のように動作します：- 今日の日付（2025年2月27日）を YYYY-MM-DD 形式（例：`2025-02-27`）で使用します。- `_posts/en` ディレクトリにファイルを作成し、ファイル名は `2025-02-27-{name}-en.md` となります。ここで `{name}` は提供された名前です（例：`test-post` は `2025-02-27-test-post-en.md` となります）。- ドラフトファイルと同じフロントマターを含めます。例：`---
audio: false
lang: en
layout: post
title: {name}
translated: false` --- `_posts/en` ディレクトリが存在することを確認し、必要に応じて作成します。

予期せぬ詳細 興味深いことに、ドラフトファイルと新しいオリジナルファイルは同じ日付ベースの命名規則を共有していますが、`delete_md` 関数は日付のないファイル（例：`{name}-en.md`）を検索しており、これはファイル管理に不一致を引き起こす可能性があります。完全な互換性を確保するために、削除ロジックを更新して日付付きのファイル名を処理する必要があるかもしれません。

アンケートノート：`create-original` コマンドの追加の詳細な分析

このセクションでは、提供された Python スクリプトに `create-original` コマンドを実装することの詳細な分析を提供し、直接の回答に基づいてスクリプトの構造、実装の理由、そして潜在的な影響についての洞察を深めます。この分析は、スクリプトの既存の機能とユーザーの要求に基づいており、ユーザーが「オリジナルディレクトリ」に同じ形式のドラフトファイルを作成する新しいコマンドを追加することを希望しています。

背景とコンテキスト スクリプトは「scripts」ディレクトリにあり、「file.py」という名前で、おそらく静的サイトジェネレーターである Jekyll を使用する多言語ブログまたはコンテンツ管理システムのための Markdown ファイルの作成と削除を処理しています。現在、3つのコマンドをサポートしています：- `create`：現在の日付を含むファイル名（例：2025-02-27-{name}-en.md）で `_drafts` ディレクトリにドラフト Markdown ファイルを作成します。- `create-note`：日付ベースのファイル名で `notes` ディレクトリにノートファイルを作成します。- `delete`：複数の言語の `_posts` ディレクトリと関連するアセットディレクトリから Markdown ファイル、PDF、オーディオファイルを削除し、日付のないファイル名（例：{name}-{lang}.md）を検索します。

ユーザーは、デフォルトのドラフト作成 (`create` コマンド) と同じ形式で「オリジナルディレクトリ」に直接ファイルを作成する `create-original` コマンドを追加することを希望しています。コンテキストに基づいて、「オリジナルディレクトリ」は `_posts/en`、英語の投稿用ディレクトリと解釈されます。

実装の詳細 ユーザーの要求を満たすために、新しい関数 `create_original` が設計され、`create_md` 関数を模倣し `_posts/en` ディレクトリをターゲットにしました。実装の詳細は以下の通りです：

- **日付の処理**：関数は `datetime.date.today()` を使用して現在の日付を取得し、2025年2月27日04:00 AM PST の場合は 2025-02-27 となります。この日付は YYYY-MM-DD 形式でドラフトファイル名と一致するようフォーマットされます。
- **ディレクトリとファイルパス**：関数は `_posts/en` ディレクトリが存在することを確認し、必要に応じて `os.makedirs` を使用して作成します。ファイルは `os.path.join('_posts', 'en', f'{date_str}-{name}-en.md')` で作成され、ファイル名には日付が含まれます（例：`test-post` の名前の場合は 2025-02-27-test-post-en.md）。
- **フロントマター**：フロントマターは `create_md` と同じで、以下のように定義されます：

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

これにより、ドラフトファイルとの一貫性が保たれ、オーディオアタッチメントがないことを示す `audio: false`、英語を示す `lang: en`、投稿タイトルを示す `title: {name}` などのフィールドが含まれます。

- **ファイルの作成**：ファイルは `open(file_path, 'w', encoding='utf-8')` を使用して作成され、広範な互換性を確保するために UTF-8 エンコーディングが使用され、確認メッセージが表示されます（例：`Created original file: _posts/en/2025-02-27-test-post-en.md`）。

スクリプトのメイン部分は `create-original` を有効なアクションとして追加し、使用メッセージを以下のように変更しました：

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

そして、`create_original(name)` を呼び出す条件を追加しました。

既存の関数との比較 以下の表は、`create_md`、`create_note`、`create_original` の違いと類似点を示しています：

| 関数名 | ディレクトリ | ファイル名形式 | フロントマターフィールド | メモ |
|------------------------------|------------------------|--------------------------------------|---|-------------------------|
| <code>create_md</code> | <code>_drafts</code> | <code>{date_str}-{name}-en.md</code> | <code>audio, lang, layout, title, translated</code> | 英語の投稿用のドラフトファイルを作成 |
| <code>create_note</code> | <code>notes</code> | <code>{date_str}-{name}-en.md</code> | <code>title, lang, layout, audio, translated</code> | ノートファイルを作成、類似のフロントマター |
| <code>create_original</code> | <code>_posts/en</code> | <code>{date_str}-{name}-en.md</code> | <code>audio, lang, layout, title, translated</code> | 新しいコマンド、ドラフトと同じ形式、投稿にある |

この表は、`create_original` が `create_md` とファイル名形式とフロントマターで一致しているが、`_posts/en` ディレクトリをターゲットにし、ドラフトステージをバイパスしていることを示しています。

潜在的な影響と考慮事項 実装はユーザーの要求を満たしていますが、特に既存の `delete_md` 関数との関係でいくつかの影響があります：- **ファイル名の不一致**：`delete_md` 関数は `_posts/lang` ディレクトリ内のファイル名 `{name}-{lang}.md` を検索しています（例：`_posts/en/test-post-en.md`）が、`create_original` は日付を含むファイル（例：`_posts/en/2025-02-27-test-post-en.md`）を作成します。この不一致により、`delete_md` は `create_original` によって作成されたファイルを見つけられない可能性があります。完全な互換性を確保するために、日付付きのファイル名を処理するように削除ロジックを更新する必要があるかもしれません。- **サイト構造**：スクリプトは `_posts` ディレクトリ内の各言語（`en`、`zh` など）のサブディレクトリを持つ多言語セットアップを示唆しており、`delete_md` のパターンに日付が含まれていないことから、`_posts` 内の投稿はファイル名の日付をソートに使用していない可能性があります。これは、Jekyll ベースのサイトではファイル名の日付が投稿の日付を決定するのが一般的ですが、スクリプトの現在の動作と一致しています。- **言語の範囲**：実装は英語（`lang: en`）に焦点を当てており、`create_md` とユーザーの要求が示唆しています。他の言語で `create-original` が必要な場合、`create_md` と同様に関数を拡張する必要があります。

理由と意思決定 「デフォルトの作成ドラフトと同じ形式」という解釈は重要でした。最初は、これはファイル名形式（日付を含む）かフロントマターかを示すかどうかが不明でした。ユーザーが「形式」に強調を置き、`create_md` の例（ファイル名に日付を含む）に基づいて、`create_original` が `{date_str}-{name}-en.md` を使用することが決定されました。しかし、この決定は `delete_md` との潜在的な競合を示唆し、ファイル管理のための将来の調整が必要になるかもしれません。

「オリジナルディレクトリ」は、`delete_md` の `_posts/lang` での操作に基づいて `_posts/en` と推測されました。この仮定は、投稿が `_posts` にあり、`/en` サブディレクトリが多言語コンテキストに適合するため、典型的なブログセットアップと一致しています。

結論 `create-original` コマンドの追加は、スクリプトの機能を向上させ、`_posts/en` に直接日付付きの Markdown ファイルを作成し、ドラフトと同じフロントマターを保持することができます。2025 年 2 月 27 日の場合、ファイルは `2025-02-27-{name}-en.md` となり、ドラフト形式と一致します。しかし、ユーザーは `delete_md` 関数とのファイル名の不一致に注意する必要があり、完全な互換性を確保するために更新が必要になるかもしれません。

主な引用

- Python `os` モジュールのドキュメント
- Python `datetime` モジュールのドキュメント
- Jekyll 投稿のドキュメント