



業務改善提案

Image2Webp

フォルダを選ぶだけ。非エンジニアにもやさしい画像変換ツール

2025年12月29日

01

背景課題・ターゲット・競合リサーチ

目的・課題

- Webサイト更新のたびに画像変換が手間
- PDFをバナーに変換するのが毎回面倒
- オンライン変換はセキュリティや手間に課題
- ローカルでシンプルに使えるツールにニーズがあると判断した。

ターゲット

- Webサイト更新担当者
- 業務資料を作成・変換する事務職
- 非エンジニアでも使えるツールが欲しい現場スタッフ

競合分析／リサーチ

類似ツールはオンライン変換サービスやPhotoshopのバッチ処理機能があるが、

- インターネットにアップロードが必要
- PDFは未対応
- UIが複雑

という課題があり、ローカルでシンプルに使えるツールにニーズがあると判断した。

02

コンセプト・構成図

コンセプト

「選ぶだけで変換完了」

エンジニアでなくても扱えることを最優先し、操作手順は最小限に。

コマンド不要、実行するとフォルダ選択だけで完結するUI設計。

構成図・要素

入力フォルダ選択

→ ファイル走査

→PDFページ分割（必要時）

→WebP変換

→出力フォルダに保存

→作成数を表示

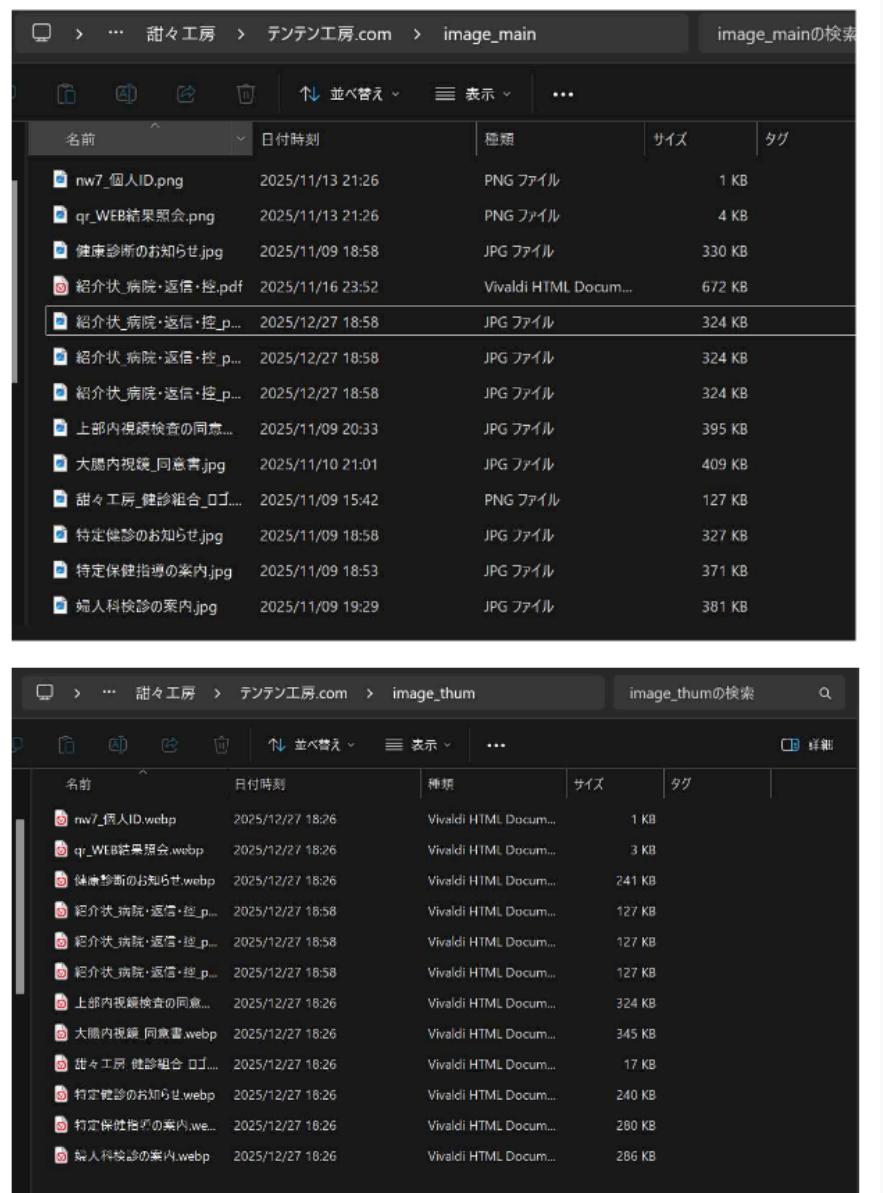
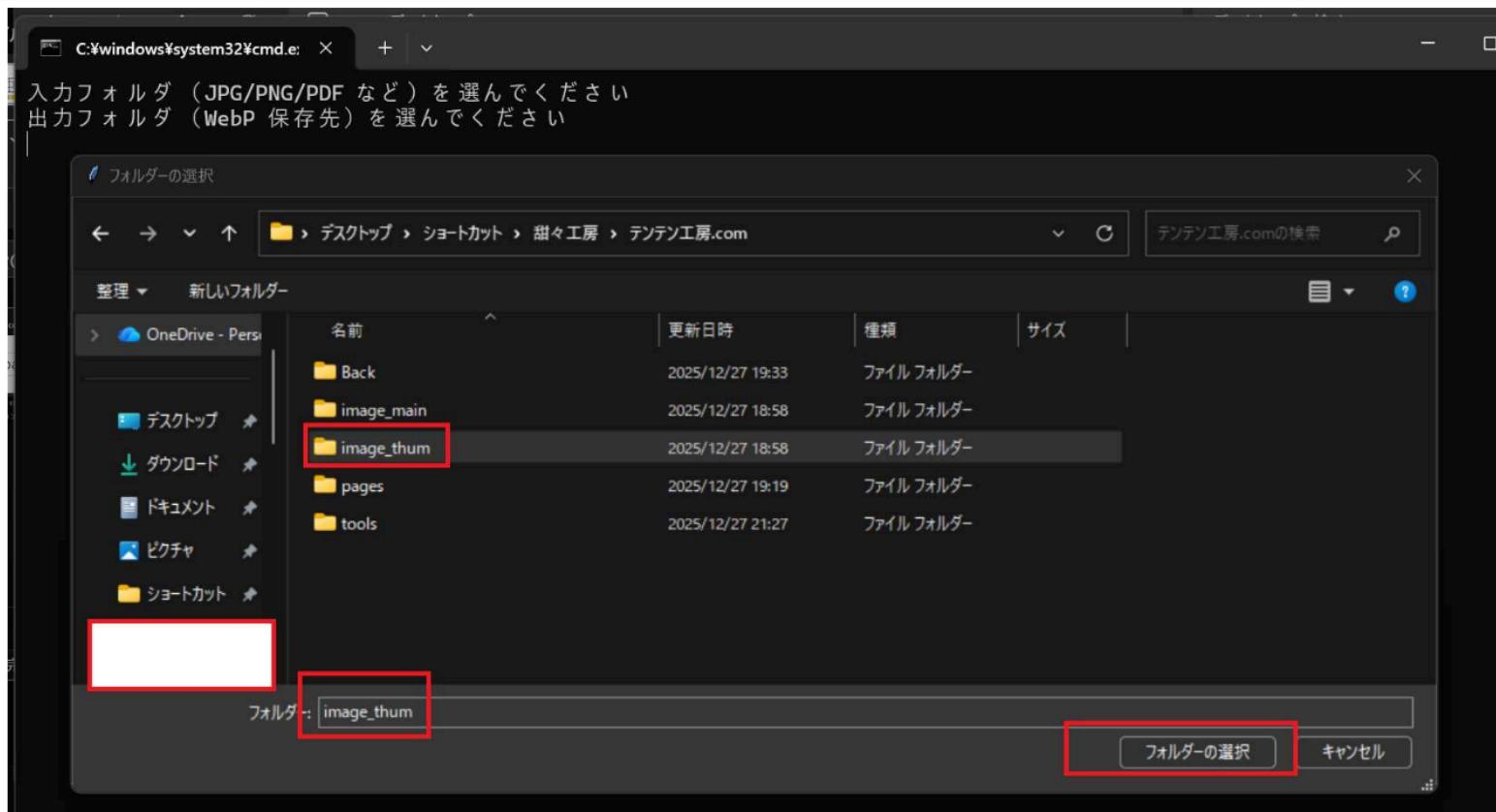
デザイン設計

- Tkinterを使ったGUI（フォルダ選択用）
- CLIにも対応可能だが、GUI選択により非エンジニア対応を重視
- ファイル名に日付や連番は付与せず、元のファイル名を尊重

画面デザイン（成果物）

- 起動時：黒背景のターミナルで「フォルダ選択」が2回表示
- 入力フォルダ → 出力フォルダ → 自動処理開始
- ステータスログがリアルタイムで表示
(例：✓ 完了 → sample.webp)

実行画面・利用シーン



- 入力フォルダ、出力フォルダの選択
- 変換後、webpファイルの作成が始まる
- WEBサイト上の画像が増えても軽い状態で使用が可能

工夫

PDFはpypdfium2で全ページをJPGに変換してからWebP化

- PDFはpypdfium2で全ページをJPGに変換してからWebP化
- WebPファイルが既に存在する場合は自動スキップ
- GUI不要の環境でも動作するよう関数分離（main実行部とロジック）

学び

- 非デザイナーや事務職でも使える「やさしいUX設計」の大切さを実感
- ファイル命名やフォルダ構成のルールを想定した設計力が向上
- PDFや画像処理のライブラリ（Pillow / pypdfium2）の実践的な扱いを習得

作業項目	内容	時間	人日換算
要件定義	使用目的・操作手順・対象拡張子の確認	2h	0.25人日
技術調査	Pillow / pypdfium2 / Tkinter の調査	4h	0.5人日
設計	変換ロジック・UI・ファイル命名ルール設計	6h	0.75人日
実装	メイン処理 / PDF対応 / ログ出力処理	12h	1.5人日
テスト・改善	エラーハンドリング / 異常系テスト	4h	0.5人日
ドキュメント作成	利用手順書・PDF出力スクリプト含む	4h	0.5人日

合計：32時間（約4人日） ※1人日=8時間想定

- **納品形態**

- ツールはPythonスクリプト形式で提供 (.py)
- 導入先PCにPython環境があれば即使用可能
- READMEまたはPDF資料付き

- **導入ハードル**

- オンライン不要（ローカル実行）
- 初期セットアップ（ライブラリ導入）に10分程度

- **ランニングコスト**

- なし（インターネット接続・サーバー費不要）
- 同様の変換要件が継続する限り、何度でも利用可能
- アップデートの必要も低く、メンテナンスフリー



Question

画像変換精度や画質は落ちませんか？

Answer

WebP変換時に画質90（高品質設定）で保存しており、
目視では劣化を感じないレベルです。

他の画像変換ツールと比べて、何が違うの？

オンライン変換と違い、ローカル完結・PDF対応・非
エンジニア向けという3点が差別化ポイントです。

このツールは今後も保守されますか？

機能要件がシンプルなため、現在の完成版で継続利用
が可能です。Pythonライブラリが互換性を維持する限
り、再構築の必要もありません。