

高度なドキュメント構造テスト

1. 概要

このドキュメントは、複雑な PDF 変換テストのためのマークダウンファイルです。様々な要素を組み合わせて、実際の業務文書に近い構造を持っています。

1.1 目的

- PDF から Markdown への変換精度の測定
- 複雑な構造の保持確認
- 多様なコンテンツタイプの処理能力評価

1.2 検証項目

1. 階層構造の保持
2. 表形式データの正確性
3. 数式・特殊記号の処理
4. 引用・コードブロックの維持

2. 技術仕様

2.1 システム要件

項目	最小要件	推奨要件	備考
OS	Windows 10 / macOS 10.15 / Ubuntu 18.04	最新版	64bit 必須
メモリ	8GB	16GB 以上	大容量ファイル処理時
ストレージ	100MB	1GB 以上	テンポラリファイル含む
ネットワーク	オフライン可	ブロードバンド	オンライン機能利用時

2.2 対応フォーマット

2.2.1 入力フォーマット

- PDF (v1.4-2.0)
 - 暗号化 PDF は非対応
 - OCR 結果テキストも処理可能
 - ファイルサイズ上限: 500MB
- DOCX (Office 2007 以降)
 - マクロ含有ファイルは無視
 - 埋め込みオブジェクトは一部制限あり

2.2.2 出力フォーマット

- Markdown (CommonMark 準拠)
- HTML (HTML5 標準)
- Plain Text (UTF-8 エンコーディング)

3. 詳細機能

3.1 テーブル処理機能

複雑なテーブル構造の処理について説明します。

3.1.1 基本テーブル

製品名	価格	在庫数	販売状況
Product A	¥1,200	150	販売中
Product B	¥2,500	45	残りわずか
Product C	¥850	0	売り切れ
Product D	¥3,100	200	販売中

3.1.2 結合セルを含むテーブル

部門	担当者	業務内容	進捗
開発部	田中太郎	フロントエンド開発	80%
	佐藤花子	バックエンド開発	65%
	鈴木一郎	データベース設計	90%
営業部	山田次郎	新規開拓	45%
	高橋三郎	既存顧客フォロー	70%

3.2 数式・記号処理

3.2.1 数学記号

- 基本演算子: $+$ $-$ \times \div $=$ \neq \approx ∞
- 比較演算子: $<$ $>$ \leq \geq
- 集合記号: \in \notin \subset \supset \cup \cap \emptyset
- ギリシャ文字: α β γ δ ϵ π σ Σ Ω

3.2.2 特殊文字

- 通貨記号: \$ € £ ¥ ₩ □
- 著作権記号: © ® ™ ®
- 矢印: \rightarrow \leftarrow \uparrow \downarrow \Rightarrow \Leftarrow \Uparrow \Downarrow
- その他: § ¶ † ‡ • ‰ ‰

3.3 コードブロック

3.3.1 プログラミング言語

```
def convert_pdf_to_markdown(file_path):  
    """  
    PDF ファイルを Markdown に変換する関数  
    """  
    try:  
        converter = AnyConverter()  
        result = converter.convert(file_path)  
  
        if result.warnings:  
            for warning in result.warnings:  
                print(f"Warning: {warning}")  
  
        return result.text  
    except Exception as e:  
        print(f"Error: {e}")  
        return None  
  
# 使用例  
markdown_text = convert_pdf_to_markdown("sample.pdf")  
print(markdown_text)
```

3.3.2 設定ファイル

```
{  
  "converter": {  
    "input_formats": ["pdf", "docx"],  
    "output_format": "markdown",  
    "options": {  
      "preserve_formatting": true,  
      "extract_images": false,  
      "table_detection": "auto",  
      "language": "ja-JP"  
    }  
  },  
  "performance": {  
    "max_file_size": "500MB",  
    "timeout_seconds": 300,  
    "parallel_processing": true  
  }  
}
```

4. 引用とリスト

4.1 複数レベルの引用

これは第一レベルの引用です。

これは第二レベルの引用です。より詳細な説明がここに含まれます。

さらに深いレベルの引用も可能です。

第一レベルに戻ります。

4.2 複雑なリスト構造

1. 第一章: 基礎編
 - 1.1 概要説明
 - 1.1.1 背景
 - 1.1.2 目的
 - 1.2 基本操作
 - 1.2.1 インストール
 - 1.2.2 初期設定
2. 第二章: 応用編
 - 2.1 高度な機能
 - ☐ 機能 A の実装
 - ☒ 機能 B の実装完了
 - ☐ 機能 C のテスト
 - 2.2 カスタマイズ
 - 設定ファイルの編集
 - プラグインの追加
3. 第三章: トラブルシューティング
 - よくある質問
 - エラー対処法

5. 複合コンテンツ

5.1 表内のマークダウン要素

機能	説明	コード例	状態
太字処理	テキストを太字に変換	**text**	<input type="checkbox"/> 実装済み
斜体処理	テキストを斜体に変換	<i>*text*</i>	<input type="checkbox"/> 実装済み
コード	インラインコード表示	<code>`code`</code>	<input type="checkbox"/> 実装済み
取り消し線	テキストに取り消し線	~~text~~	<input type="checkbox"/> 開発中

5.2 ネストしたコンテンツ

重要な注意事項

この機能を使用する前に、以下の点を確認してください：

1. システム要件を満たしているか
2. 必要な権限が設定されているか
3. バックアップが取得されているか

事前チェックコマンド

```
system-check --requirements
permission-check --user current
backup-status --verify
```

チェック項目	確認方法	結果
システム要件	system-check	<input type="checkbox"/> 要確認
権限設定	permission-check	<input type="checkbox"/> OK
バックアップ	backup-status	<input type="checkbox"/> 古い

6. 結論

このドキュメントは、PDF から Markdown への変換テストにおいて、以下の要素を検証するために作成されました：

- 構造の複雑性: 多階層見出し、ネストしたリスト
- データの多様性: テーブル、コードブロック、引用
- 記号の処理: 特殊文字、数学記号、絵文字
- フォーマットの保持: 太字、斜体、取り消し線

6.1 期待される結果

変換後の Markdown ファイルにおいて、以下が適切に保持されることを期待します：

1. ☐ 見出し階層の維持
2. ☐ テーブル構造の保持
3. ☐ 特殊記号の一部変換
4. ☐ 複雑なネスト構造の簡略化

注記: このテストケースは、実際の業務文書に近い複雑さを持つよう設計されています。