PDF/Officeファイルの画像変換スクリプトV1.0

目次

[一、 ダイジェスト 3](#_Toc181279923)

[二、 必要なライブラリ 4](#_Toc181279924)

[三、 PDF to Image Converter 5](#_Toc181279925)

[**1.** **必要条件** 5](#_Toc181279926)

[**2.** **使用方法** 5](#_Toc181279927)

[**3.** **出力** 5](#_Toc181279928)

[**4.** **サンプルの参照** 5](#_Toc181279929)

[四、 Word to PDF Converter 5](#_Toc181279930)

[**1.** **必要条件** 5](#_Toc181279931)

[**2.** **使用方法** 6](#_Toc181279932)

[**3.** **出力** 6](#_Toc181279933)

[**4.** **注意事項** 6](#_Toc181279934)

[**5.** **サンプルの参照** 6](#_Toc181279935)

[五、 Excel　to PDF Converter 6](#_Toc181279936)

[**1.** **必要条件** 6](#_Toc181279937)

[**2.** **使用方法** 6](#_Toc181279938)

[**3.** **出力** 7](#_Toc181279939)

[**4.** **注意事項** 7](#_Toc181279940)

[**5.** **サンプルの参照** 7](#_Toc181279941)

[六、 PPT to PDF Converter 7](#_Toc181279942)

[**1.** **必要条件** 7](#_Toc181279943)

[**2.** **使用方法** 7](#_Toc181279944)

[**3.** **出力** 8](#_Toc181279945)

[**4.** **注意事項** 8](#_Toc181279946)

[**5.** **サンプルの参照** 8](#_Toc181279947)

[七、 付録：スクリプトの出力サンプル 9](#_Toc181279948)

[**1.** **PDF to Image Converter** 9](#_Toc181279949)

[**2.** **Word to PDF Converter** 10](#_Toc181279950)

[**3.** **Excel　to PDF Converter** 11](#_Toc181279951)

[**4.** **PPT　to PDF Converter** 13](#_Toc181279952)

1. ダイジェスト

このプロジェクトには、以下の4つの主要なスクリプトが含まれている。それぞれ特定の機能を提供しており、システム資源の使用を最小限に抑え、簡単にインポートして使用できる設計となっている。

* **PDF画像変換スクリプト (pdf\_to\_image\_converter.py):**

このスクリプトは、PDFファイルを高解像度の画像に変換する。各ページは個別の画像ファイルとして保存され、画像のサイズや解像度を調整するオプションも提供されている。プロセスはメモリ内で行われるため、システムリソースの消費が少なく、効率的な処理が可能である。

* **WordからPDFへの変換スクリプト (word\_to\_pdf\_converter.py):**

このスクリプトは、Word文書（.docx）をHTML形式に変換し、その後、HTMLをPDF形式に変換する。HTML変換にはテキストの整形やスタイルの適用が行われ、PDFではカスタムフォントが使用される。全体のプロセスはメモリ内で完結し、ディスクI/Oを最小限に抑えている。

* **ExcelからPDFへの変換スクリプト (excel\_to\_pdf\_converter.py):**

このスクリプトは、PowerPointプレゼンテーション（PPTXファイル）をPDF形式に変換する。プレゼンテーションをHTMLに変換し、それをPDFに変換する機能を持つ。また、日本語フォントに対応しており、テキストやテーブルを適切に処理する。変換プロセスは効率的で、システムの資源を無駄にしない設計になっている。

* **PPTからPDFへの変換スクリプト (ppt\_to\_pdf\_converter.py)**

このスクリプトは、PowerPointプレゼンテーション（PPTXファイル）をPDF形式に変換する。プレゼンテーションをHTMLに変換し、それをPDFに変換する機能を持つ。また、日本語フォントに対応しており、テキストや画像を簡単に処理する。変換プロセスは効率的で、システムの資源を無駄にしない設計になっている。

これらのスクリプトは軽量であり、どのようなPythonプロジェクトにも簡単に統合することが可能である。必要に応じてモジュールをインポートし、簡単に機能を呼び出すことができる。システムリソースの使用が低いため、大規模なデータ処理でも安心して使用できる。

一方、フォーマットの復旧は強みではないため、大規模かつシンプルなOfficeファイルを変換する場合に向いている。

1. 必要なライブラリ

以下は、本デモに必要なライブラリの説明を示したものである。

* **openpyxl:**

Excelファイル（.xlsx）の読み書きを行うためのライブラリである。これにより、Excelデータを取得して処理できる。

* **reportlab:**

PDFを生成するための強力なライブラリである。ページサイズの設定やフォントの追加、PDFコンテンツの描画が可能である。

* **fitz (PyMuPDF):**

PDFファイルを操作するためのライブラリである。PDFからのテキスト抽出や画像の読み込み・生成が可能である。

* **PIL (Pillow):**

画像処理を行うためのライブラリである。画像の読み込み、保存、加工に使用される。

* **mammoth:**

Word文書（.docx）からHTMLへの変換を行うライブラリである。HTML形式で内容を取得するために使用される。

* **textwrap:**

テキストの折り返しや整形を行うための標準ライブラリである。PDF生成時のテキスト処理に使用される。

* **BeautifulSoup:**

HTMLやXMLのパースを行うライブラリである。HTML文書から特定の要素を取得するために使用される。

* **xhtml2pdf:**

HTMLをPDFに変換するためのライブラリである。ウェブ技術を利用して高品質なPDFを生成できる。

* **python-pptx:**

PowerPointプレゼンテーションを操作するためのライブラリである。スライドの読み込みや編集が可能であり、PPTファイルをPDFに変換する際に利用される。特に、日本語フォントのサポートやスライド内の画像やテキストの処理において重要である。

これらのライブラリは、各スクリプトの機能を実現するために必要不可欠であり、プロジェクトに含まれていることが重要である。それぞれのライブラリが持つ特性を理解して、適切に使用することが求められる。

1. PDF to Image Converter

このスクリプトは、PDFファイルの各ページを高解像度の画像に変換し、指定されたフォルダに保存するためのクラスPDFToImageConverterを提供するものである。このクラスは、処理するPDFファイルのパスと出力画像を保存するベースフォルダを引数として受け取る。

1. **必要条件**

以下のライブラリが必要である。これらがインストールされていない場合、事前にインストールされたい。

* **fitz**
* **PIL**

1. **使用方法**
2. **初期化**: PDFToImageConverterクラスのインスタンスを作成する。コンストラクタには、入力PDFファイルのパスと出力フォルダのベースパスを指定する。

**converter = PDFToImageConverter(input\_path='path/to/your/file.pdf', output\_base\_folder='path/to/output/folder')**

1. **変換**: convert\_pdf\_to\_imagesメソッドを呼び出して、PDFを画像に変換する。オプションで、画像の解像度を調整するためのズームファクターを指定できる（デフォルトは2.0）。

**converter.convert\_pdf\_to\_images(zoom\_factor=2.0)**

1. **出力**

変換された画像は、指定した出力フォルダ内にpage\_1.png, page\_2.png, ...の形式で保存される。すべてのページが保存された後、完了メッセージが表示される。

1. **サンプルの参照**

このスクリプトの出力サンプルは[付録1](#付録１)にて参照される。

1. Word to PDF Converter

このスクリプトは、Word文書（.docxファイル）をHTMLに変換し、加工した後にPDF形式で保存するためのクラスWordToPDFConverterを提供するものである。これにより、簡単にWordファイルを高品質なPDFに変換することができる。

1. **必要条件**

このスクリプトの実行には、以下のPythonライブラリが必要である。これらがインストールされていない場合、事前にインストールされたい。

* **mammoth**
* **BeautifulSoup（bs4）**
* **xhtml2pdf**
* **reportlab**
* **textwrap**

1. **使用方法**
2. **初期化:** WordToPDFConverterクラスのインスタンスを作成する。コンストラクタには、入力となるWordファイルのパス、出力先のPDFファイルのパス、カスタムフォントファイルのパスを指定する。

**converter = WordToPDFConverter(**

**input\_file\_path='path/to/your/document.docx', pdf\_file\_path='path/to/output/document.pdf',**

**font\_path='path/to/font.ttf' )**

1. **変換**: convert\_word\_to\_pdfメソッドを呼び出して、WordファイルをPDFに変換する。

**converter.convert\_word\_to\_pdf()**

1. **出力**

PDFは指定されたパスに保存される。変換が成功すると、その旨がコンソールに表示される。

1. **注意事項**

フォントの登録は必須で、指定されたフォントファイルが利用可能である必要がある。

1. **サンプルの参照**

このスクリプトの出力サンプルは[付録２](#付録２)にて参照される。

1. Excel　to PDF Converter

このスクリプトは、ExcelファイルをPDF形式に変換するためのクラスExcelToPDFConverterを提供するものである。指定されたフォントを使用してExcelの内容をPDFに格納し、セルのボーダーやテキストの整形をサポートする。

1. **必要条件**

このスクリプトの実行には、以下のPythonライブラリが必要である。これらがインストールされていない場合、事前にインストールされたい。

* **openpyxl**
* **reportlab**

1. **使用方法**
2. **初期化:** ExcelToPDFConverterクラスのインスタンスを作成する。コンストラクタには、日本語フォントのファイルパスを指定する。

**converter = ExcelToPDFConverter(font\_path='path/to/japanese/font.ttf')**

1. **変換:** convert\_excel\_to\_pdfメソッドを呼び出して、ExcelファイルをPDFに変換する。Excelファイルのパスと出力先のPDFファイルのパスを指定する。

**converter.convert\_excel\_to\_pdf(excel\_file='path/to/excel.xlsx', pdf\_file='path/to/output.pdf')**

1. **出力**

PDFは指定されたパスに保存される。変換が成功すると、コンソールに完了の通知が表示される。

1. **注意事項**

日本語フォントの登録は必須であり、指定されたフォントファイルが利用可能である必要がある。

セルのデータやボーダーが複雑な場合や、コラムが多すぎる場合には、PDFに正しく反映されないことがある。この場合、レイアウトやデザインの編集が必要になることがある。

出力できるのはアクティブシートのみであるため、出力したいシートをアクティブにする必要がある。

1. **サンプルの参照**

このスクリプトの出力サンプルは[付録３](#付録３)にて参照される。

1. PPT to PDF Converter

このスクリプトは、PowerPointプレゼンテーション（PPTXファイル）をPDF形式に変換するためのクラスPPTToPDFConverterを提供する。プレゼンテーションをHTMLに変換し、それをPDFに変換する機能を兼ね備えている。また、日本語フォントのサポートも含まれている。

1. **必要条件**

このスクリプトの実行には、以下のPythonライブラリが必要である。これらがインストールされていない場合、事前にインストールすることが推奨される。

* **python-pptx**
* **Pillow**
* **xhtml2pdf**
* **reportlab**

1. **使用方法**
2. **初期化:** PPTToPDFConverterクラスのインスタンスを作成する際に、PPTXファイルのパス、出力PDFファイルのパス、および日本語フォントのファイルパスを指定する。

**converter = PPTToPDFConverter(pptx\_file\_path='path/to/presentation.pptx', output\_pdf\_path='path/to/output.pdf', font\_path='path/to/font.ttf')**

1. **変換:** convert\_ppt\_to\_pdfメソッドを呼び出すことで、PPTXファイルをPDFに変換する。

**converter.convert\_ppt\_to\_pdf()**

1. **出力**

変換が完了すると、指定された出力パスにPDFファイルが保存される。変換が成功した場合、完了のメッセージがコンソールに表示される。

1. **注意事項**

日本語フォントの登録は必須であり、指定されたフォントファイルが存在する必要がある。

PPTXファイルのスライドに含まれる内容が複雑な場合や特定のフォーマットが使用されている場合、PDF出力において一部の内容が正しく表示されない可能性がある。従って、シンプルなPPTファイルを推奨する。

1. **サンプルの参照**

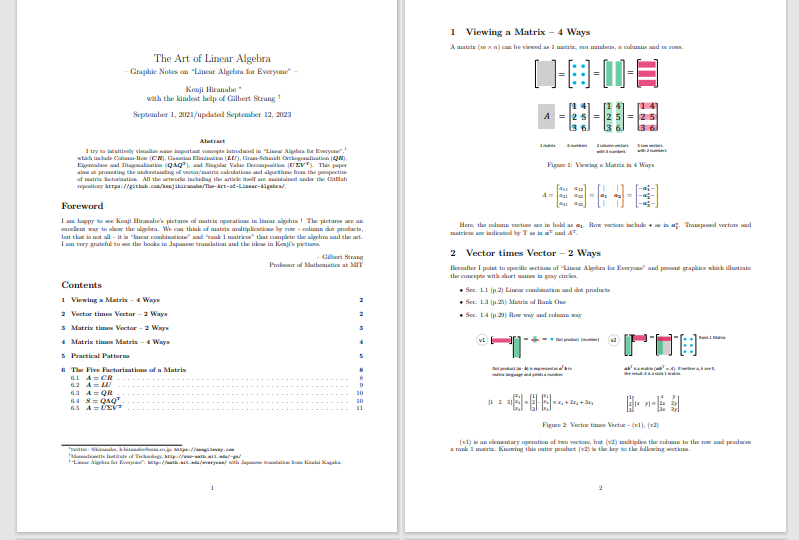
このスクリプトの出力サンプルは[付録４](#付録４)にて参照される。

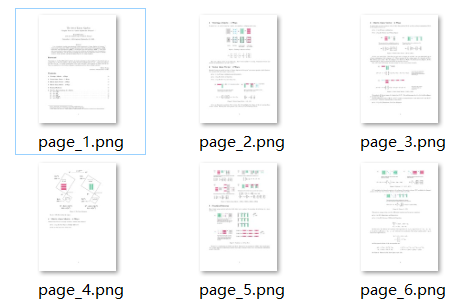
1. 付録：スクリプトの出力サンプル

この付録には、以上で紹介した4つのスクリプトのサンプルが含まれている。それぞれのテストケースと出力の結果が画像キャプチャーで展示される。

1. **[PDF to Image Converter](#本文１)**

テストケースは1つ用意した。上がPDFであり、下が出力された結果である。



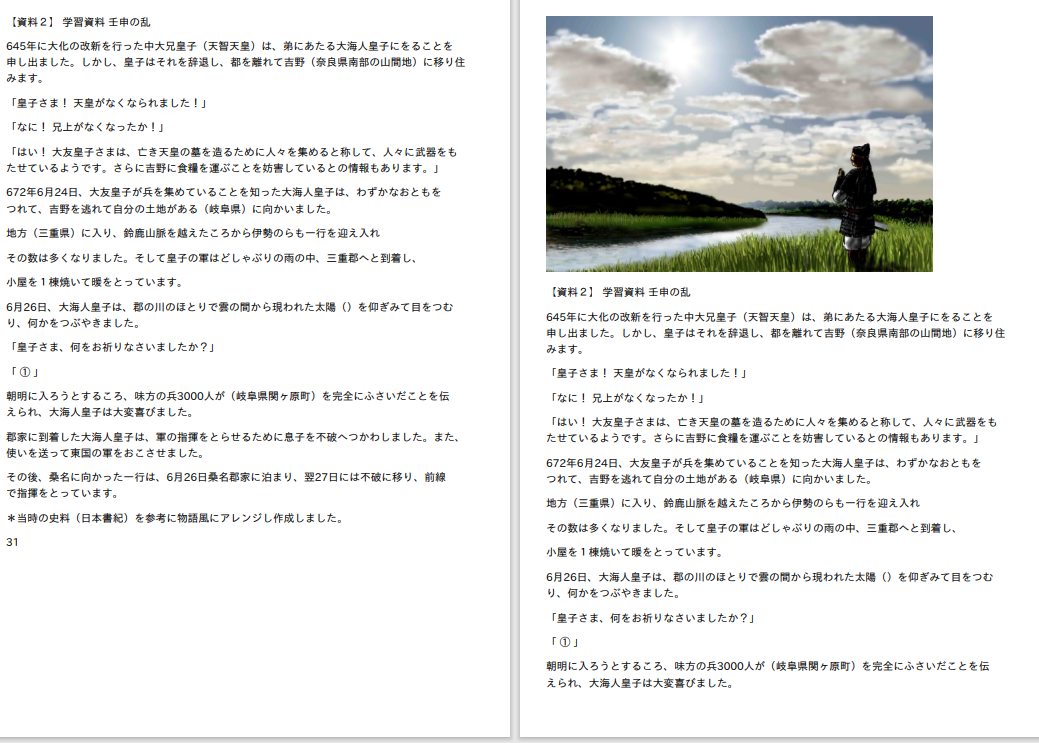


**テストケース１：簡単なPDFサンプル**

1. **[Word to PDF Converter](#本文２)**

テストケースは１つ用意した。上がWordであり、下が出力された結果である。

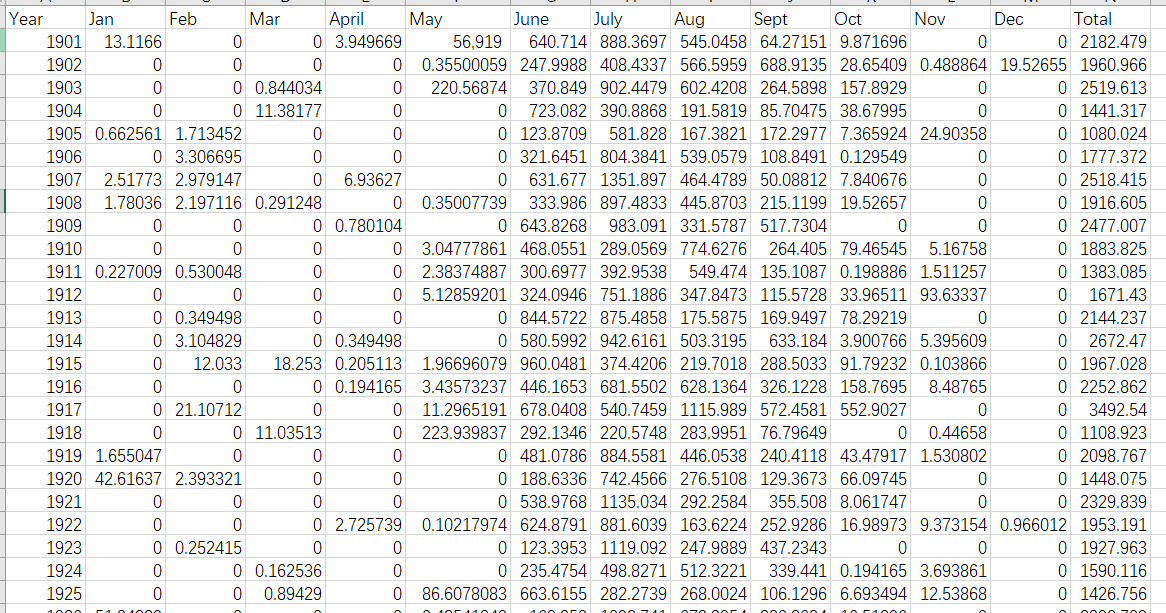


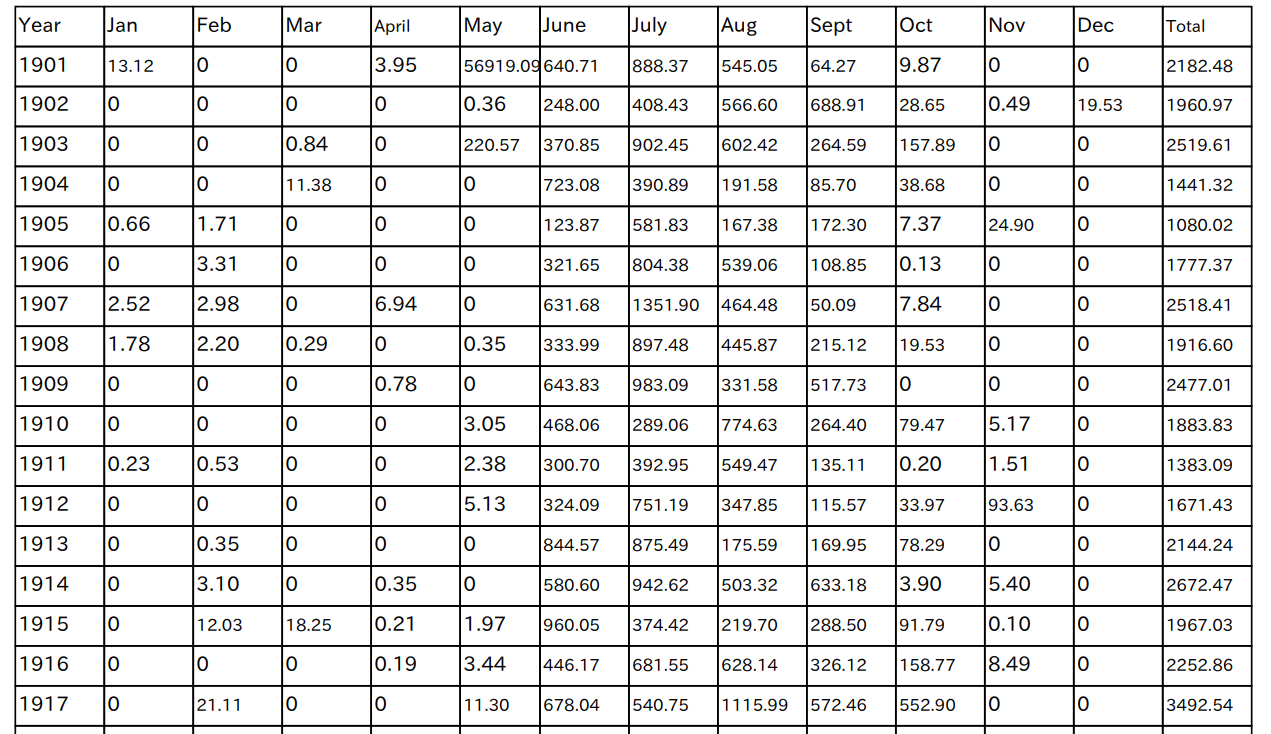


**テストケース１：文字、画像、テーブルを含むWordサンプル**

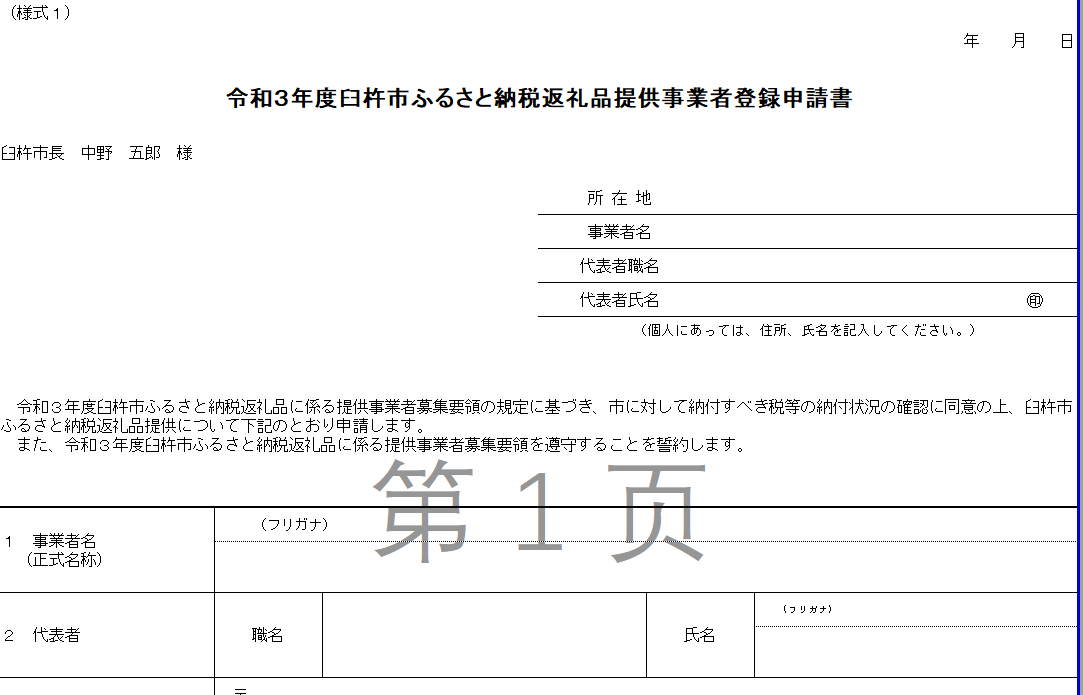
1. **[Excel　to PDF Converter](#本文３)**

テストケースは１つ用意した。上がWordであり、下が出力された結果である。





**テストケース１：データベースタイプのExcelサンプル**



**テストケース２：文書タイプのExcelサンプル**

1. **[PPT　to PDF Converter](#本文４)**

テストケースは１つ用意した。上がPPTであり、下が出力された結果である。



**テストケース１：テキスト、図及びテーブルを含むPPTサンプル**