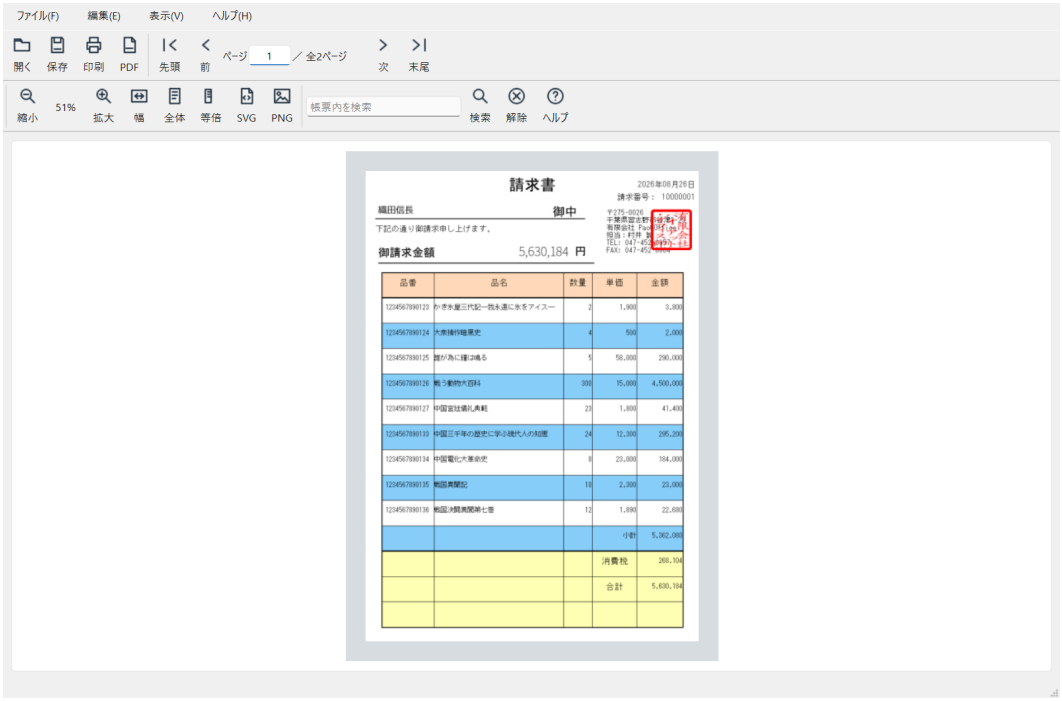
Pure Python プレビュアー 操作ガイド

Pure Pythonプレビュアーは、帳票定義や印刷データをパソコン上で開き、ページを確認し、検索・保存・印刷まで行えるデスクトップツールです。



目次

1. まず帳票を開く 3

2. ページを移動する 3

3. 拡大・縮小とスクロール 3

4. 帳票内を検索する 4

5. 保存して、もう一度開く 4

6. PDF・SVG・PNG 4

7. 印刷 4

8. 表示設定 5

9. 処理中の中止と安全な切り替え 5

10. 外部アプリへの組み込みと全UI非表示 5

    Webアプリでプレビューだけを表示する場合 6

11. ヘルプ 6

1. まず帳票を開く

Windowsの同梱ツール `Tools/Windows/Previewer/ReportsPythonSamplesAndPreview.exe` を単独で起動すると、最初はサンプル選択画面が開きます。帳票を選び「プレビュー」を押すと、以下で説明するプレビュアーが開きます。macOS・Linuxの起動方法は `Tools` 内の各OSの案内をご覧ください。

プレビュアーの「ファイル」メニューで、開くファイルの種類を選びます。

- 「定義を開く…」：`.prepd` / `.prepdj`
- 「データを開く…」：`.prepe` / `.prepej`

01で出力した帳票なら、ツールバーの「開く」または「データを開く…」から `samples/01.`あっという間に帳票出力`/output/quick-report.prepej` を選びます。デザインだけの定義ファイルは「定義を開く…」を使います。

種類を明示すれば `.xml` / `.json` も扱えます。読込に失敗しても、いま表示している帳票は残ります。

印刷データは、エクスプローラー等からプレビュアーへ1ファイルをドラッグ＆ドロップして開くこともできます。ドロップ時のカーソルはコピー操作となり、対応しないファイルや複数ファイルは受け付けません。

2. ページを移動する

先頭、前、次、末尾のボタン、またはページ番号入力を使います。番号を入力したらEnterで確定します。

- `Home` / `End`：先頭 / 末尾
- `PageUp` / `PageDown`：前 / 次

必要なときだけ「表示」メニューのページ一覧を開けます。多数ページでも全ページ分の操作部品を常時並べないため、表示は重くなりにくい構成です。

3. 拡大・縮小とスクロール

「縮小」「拡大」「幅」「全体」「等倍」を利用できます。

- `Ctrl` + マウスホイール：ポインター付近を保って拡大・縮小
- マウスホイール：上下スクロール
- `Shift` + マウスホイール：左右スクロール
- `Ctrl++` / `Ctrl+-`：拡大 / 縮小

幅または全体表示を選んだ場合、ウィンドウサイズの変更に合わせて倍率を再計算します。

4. 帳票内を検索する

検索欄へ文字を入力し、Enterまたは「検索・次候補」を押します。該当ページへ移動し、一致した文字だけを強調表示します。

- EnterまたはF3：次候補
- Shift+F3または「前候補」：前候補
- 「解除」またはEsc：検索文字と強調表示を解除
- `Ctrl+F`：検索欄へ移動

検索語を変更すると先頭候補から始まり、最後の候補から次へ進むと先頭へ戻ります。

5. 保存して、もう一度開く

帳票定義を開いている場合は、「定義を名前を付けて保存」で別名保存できます。

ツールバーの「保存」または「印刷データを名前を付けて保存…」で、表示中の完成した帳票を `.prepej` として保存できます。02で生成した帳票も、ファイルやREST APIから受け取った帳票も対象です。複数ページと画像を含めて保存するため、元のDBや画像ファイルがなくても再表示できます。

保存した `.prepej` を「開く」「データを開く…」またはドラッグ＆ドロップで開き直し、同じページと内容が表示されることを確認してください。元の帳票定義を編集用に保存する場合は、「定義を名前を付けて保存…」を使います。

6. PDF・SVG・PNG

- PDF：全ページを1つのPDFに保存
- SVG：表示対象をベクター形式で保存
- PNG：現在ページを画像として保存

画面プレビュー、PDF、SVG、PNGは同じ帳票定義とデータを元に生成されます。ただし、各形式の描画方式や利用フォント、PDF閲覧ソフトによって見え方に差が生じる場合があります。

PDFに使えるフォントに字形のない文字（絵文字など）があると、文字を黙って消さずにエラーにし、PDFは作りません。プレビューでは、その文字が□で表示されることがあります。Linuxでは日本語フォントがないとPDFを出力できません。Noto Sans CJK JPなどの日本語フォントをインストールしてください。

7. 印刷

「印刷」または `Ctrl+P` で印刷ダイアログを開きます。プリンター、部数、全ページ・現在ページ・範囲指定などを確認して実行してください。キャンセルした場合は帳票とページ位置を保ちます。

印刷完了の表示は印刷デバイスへデータを渡したことを示します。実際の用紙への排紙は、プリンター側でも確認してください。

8. 表示設定

「表示」メニューから次を切り替えられます。

- 補助ツールバーの表示
- ステータスバーの表示
- ページ一覧の表示
- ツールバー文字の表示
- 16px / 32pxのアイコンサイズ

設定は次回起動時にも利用されます。ただし、ページ移動や印刷が消えたように見えないよう、主要ツールバーの文字と補助ツールバーは起動時に表示されます。

9. 処理中の中止と安全な切り替え

時間のかかる生成中は進捗と中止操作が表示されます。中止や読込失敗で、途中まで生成した帳票へ表示を切り替えることはありません。新しい帳票の全形式が用意できてから画面を切り替えます。

10. 外部アプリへの組み込みと全UI非表示

現行版では、`PreviewWindow` のページ移動・倍率・検索・保存・印刷などのメソッドをPythonコードから呼び出せます。そのため、独自アプリからプレビュー画面を開いて操作を連携することは可能です。

配布サンプルに同梱した `PreviewWindow` では、帳票面以外のツールバー（設定ボタンを含む）、メニュー、ステータスバー、ページ一覧を一括で非表示にし、再表示できます。

```
preview.set_preview_chrome_visible(False)  # 帳票面だけを表示
preview.set_preview_chrome_visible(True)   # すべて再表示
```

これはPure Python SDK本体が提供するUI APIではなく、`02.全帳票をパソコンでプレビュー` と `04.Web API・デスクトップ連携/Client` に同梱したサンプル `PreviewWindow` の公開メソッドです。

一方、WASM版にある次の共通機能は、現行Pure Pythonデスクトップ版にはまだ揃っていません。

- 全コマンドを対象にした個別の表示設定
- SDKの安定した公開APIとしての全操作・状態取得・イベント購読

既存の「補助ツールバーを非表示」は補助ツールバーだけを切り替えます。帳票面だけにする場合は、上記の `set_preview_chrome_visible` を使用してください。

Webアプリでプレビューだけを表示する場合

03 Webサンプルと公開デモでは、プレビュアーの外側に「ツールバーを表示／非表示」ボタンがあります。初期表示は帳票だけで、ボタンを押すとPDF・検索・設定などを表示します。再表示用ボタンは外側に残るため、設定ボタンまで隠しても戻せます。

コード例と、非表示中のPDF・検索・ページ移動のAPIは、プログラマーズガイドを参照してください。Web用APIはデスクトップの `PreviewWindow` とは別です。

11. ヘルプ

F1または「ヘルプ」メニューから使い方を開けます。「バージョン情報」ではPure Pythonプレビュアーと対応出力形式を確認できます。

Pythonアプリケーションへの組み込み、PDF・SVG・PNG出力、PREPEJの読み書きは Pure Python プログラマーズガイド を参照してください。帳票を編集する場合は Pure Python デザイナー操作ガイド を参照してください。