

# pywin32 を使った Excel 操作

---

# pywin32

Win32 API にアクセスするためのパッケージです。

- openpyxl との違い -

## openpyxl

- Excel で扱えるファイル (.xlsx 等) を読み書きする。

## pywin32

- Win32 API を通して Excel 等のアプリを操作する。
- 編集集中の未保存のブックを操作することもできる。

# やりたかったこと

- 編集集中のブックに大きな画像を挿入する。
- 解像度が落ちないように自動で分割する。(\*1)
- 出来る限り少ないオペレーションで。

\*1) Excel が勝手に解像度を落とすので。

# Excel アプリの取得

```
import win32com.client  
  
xl_app = win32com.client.GetObject(  
    Class='Excel.Application')
```

- Excel が起動していないと例外が発生する。

# アクティブなブックを取得

```
wb = xl_app.ActiveWorkbook
```

- VBA の `ActiveWorkbook` プロパティに相当。
- 存在しなければ `None` となる。
- 同様にアクティブなシートやセルも取得可能。

# シートに画像を挿入

```
sheet = xl_app.ActiveSheet
sheet.Shapes.AddPicture(
    Filename=img_path,
    LinkToFile=False,
    SaveWithDocument=True,
    Left=0, Top=0, Width=-1, Height=-1
)
```

参考 : [Shapes.AddPicture メソッド \(Excel\) | Microsoft Docs](#)

# セルの座標を取得

セルに合わせて画像を貼る場合は座標を取得します。

```
cell = xl_app.ActiveCell  
x, y = cell.Left, cell.Top
```

- これを先ほどの `Left, Top` に渡す。
- 単位はポイント (1/72インチ)。

# 作ったツールの動作

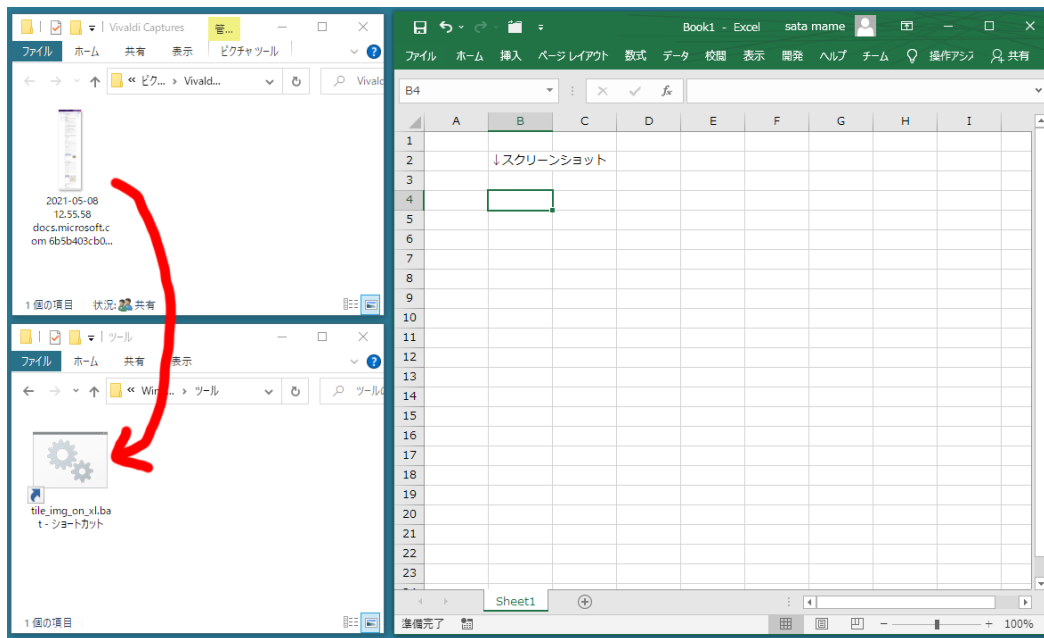
- config で分割の基準や縮尺を設定可能。(\*1)
- .bat を挟むことでドラッグアンドドロップに対応。
- PIL で画像を分割して一時ファイルとして保存。
- アクティブなシートに貼ってグループ化。

\*1) 縮尺も自動的に決まって予測できないので。



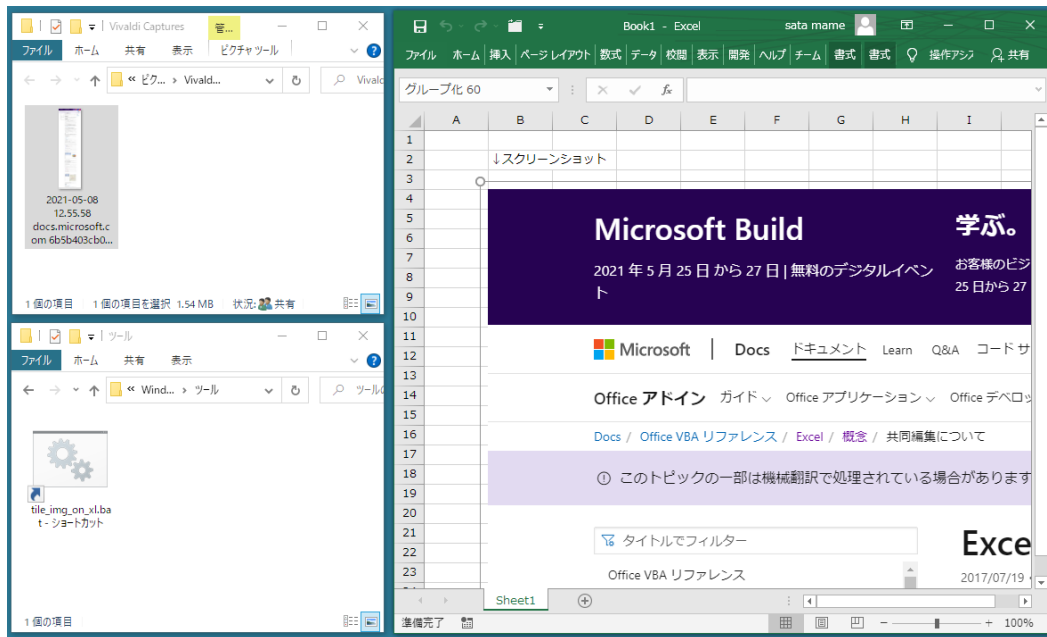
# 動作イメージ

(1) Excel を開いておいて画像 (1838x6522) を .bat に DnD。



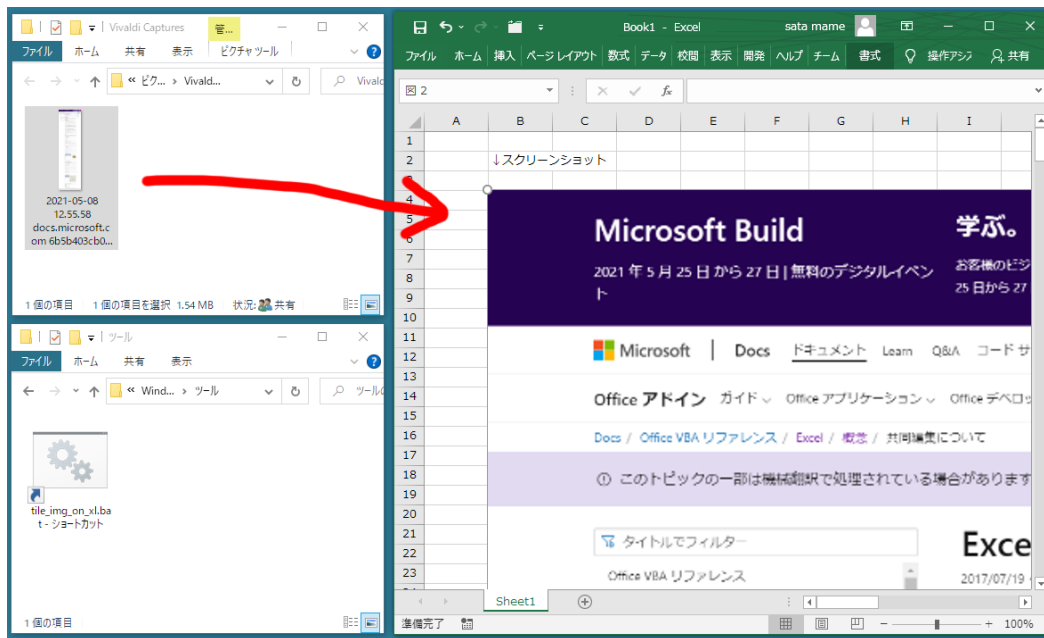
# 動作イメージ

(2) 分割・グループ化されたものがシートに貼られる。



# 動作イメージ

※ツールを使わずに画像を挿入した場合 (解像度が低い)。



実装の詳細についてはソースを御覧ください。

[GitHub : satamame/tile\\_img\\_on\\_xl](https://github.com/satamame/tile_img_on_xl)