サイン・オン

スクリプト実行アクション のすすめ

ユーザー ・・・ Zero Sys
Twitter ・・・・ https://twitter.com/sys_zero
Qiita ・・・・・ https://qiita.com/ sys_zero

Point

スクリプト実行アクション良いです！

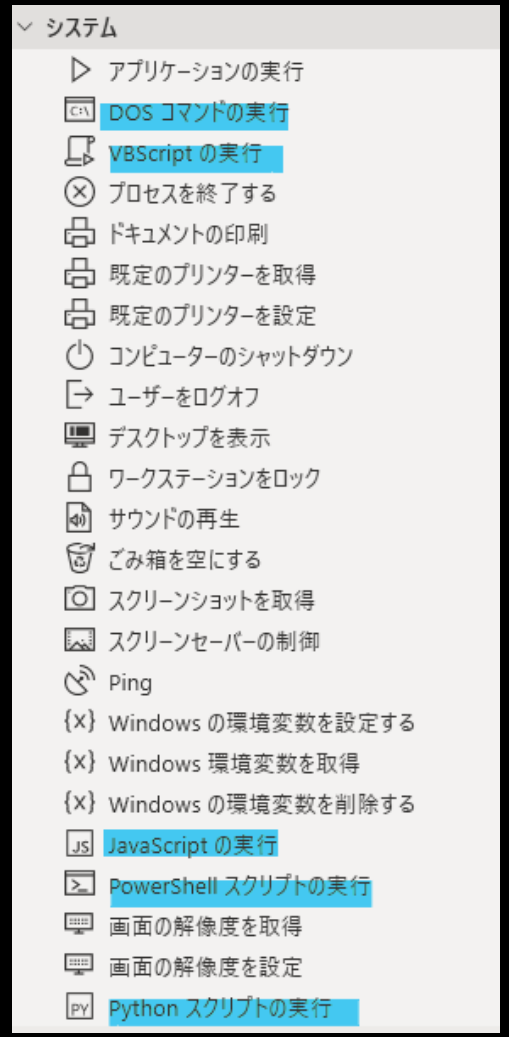
前提

- 今回のLTではガバナンスの視点はありません。
- 個人学習の範囲となります。
- 2021年7月の情報です。

スクリプト実行アクションの種類

- DOSコマンドの実行
- VBScriptの実行
- JavaScript実行
- PowerShellスクリプトの実行
- Pythonスクリプトの実行

- WebページでJavaScript関数を実行
- SQLステートメントの実行

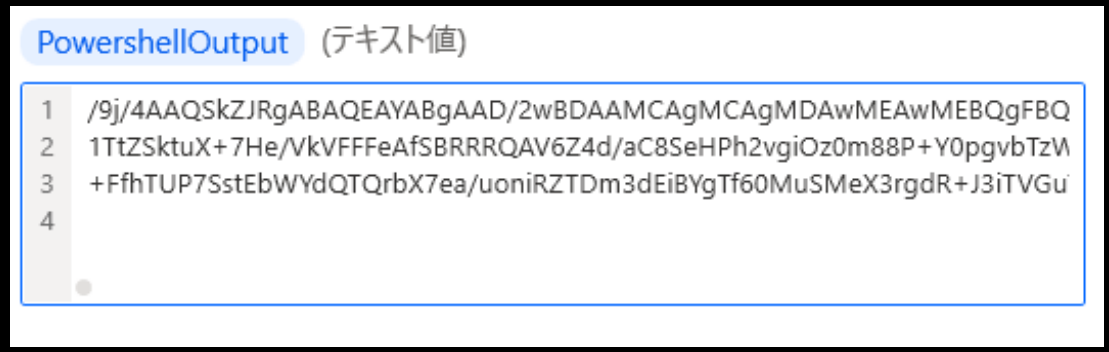


理由 1

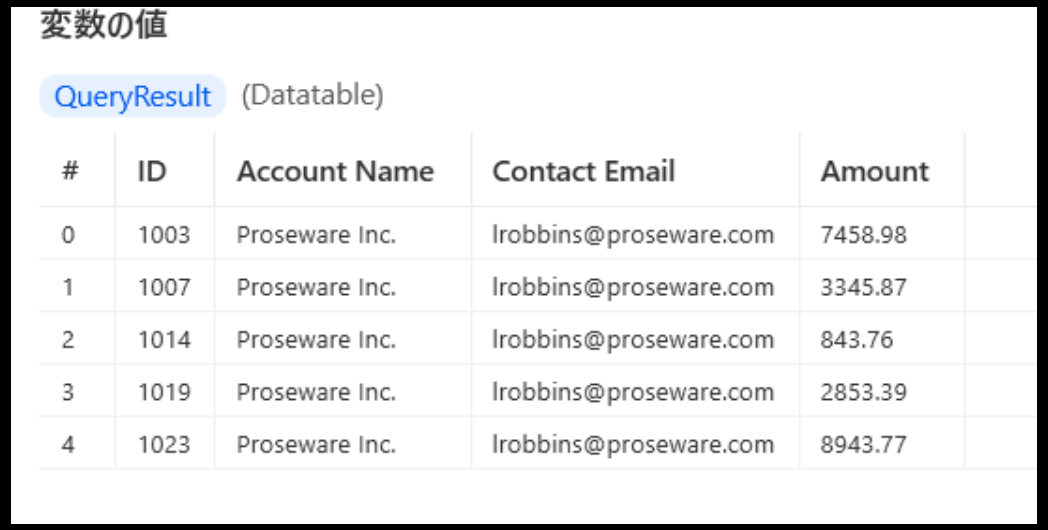
基本アクションだけでは難しいことがある

例1

画像のBase64エンコード



データテーブルの部分検索



#	ID	Account Name	Contact Email	Amount
0	1003	Proseware Inc.	lrobbins@proseware.com	7458.98
1	1007	Proseware Inc.	lrobbins@proseware.com	3345.87
2	1014	Proseware Inc.	lrobbins@proseware.com	843.76
3	1019	Proseware Inc.	lrobbins@proseware.com	2853.39
4	1023	Proseware Inc.	lrobbins@proseware.com	8943.77

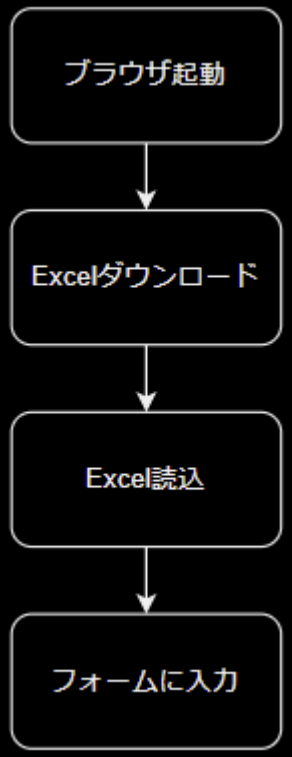
Sample data Microsoft RPA in A DAY Contoso Invoices.xlsx

理由 2

基本アクションだけで構成するより高速な場合がある

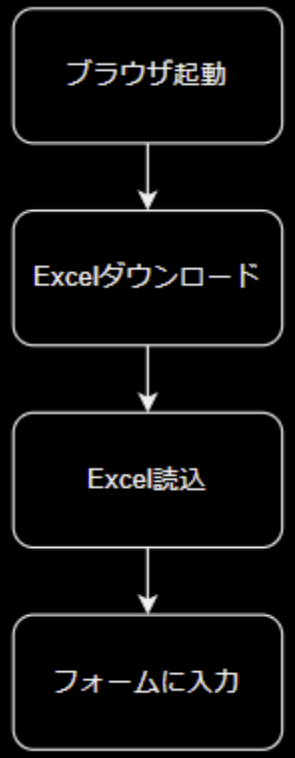
例2

http://www.rpachallenge.com/



例2

http://www.rpachallenge.com/



WebページでJavaScript関数を実行
SQLステートメントの実行

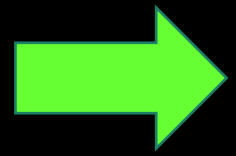
スクリプト実行アクションを推す

理由 3

全部コードで書くよりも楽なケースがある

例3

画像からの文字を消す処理を自動化してみる



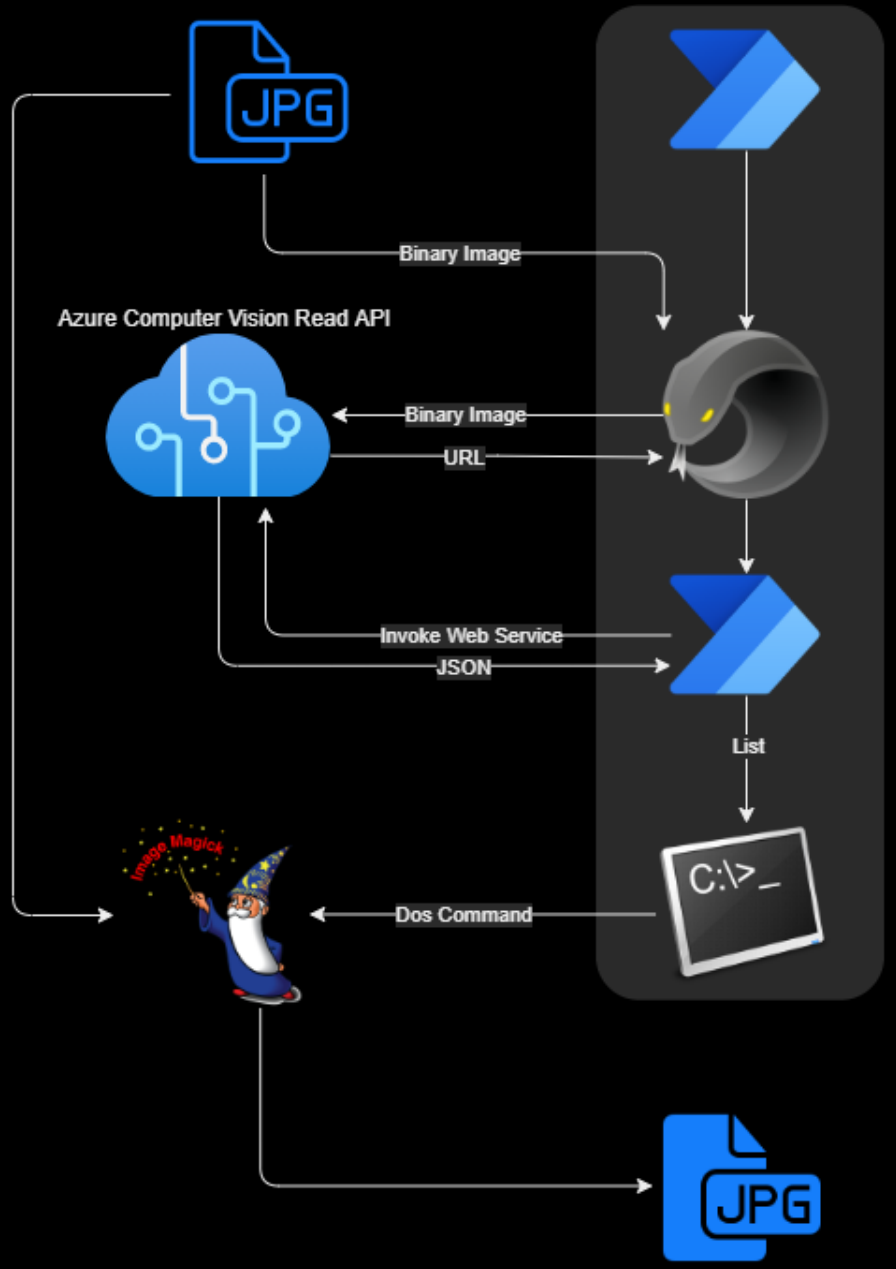
Before

After

https://www.pakutaso.com/20200948245post-30590.html

構成

Windows10 Pro 21H1
Power Automate Desktop 2.10.36.1161
Microsoft Azure Cognitive Services Computer Vision v3.2 Read API
ImageMagick 7.1.0-2 Q16 x64



恥の多い生涯を送って来ました。自分には、人間の生活というものが、見当つかないのです。自分は東北の田舎に生れましたので、汽車をはじめて見たのは、よほど大きくなってからでした。自分は停車場のブリッジを、上って、降りて、そうしてそれが線路をまたぎ越えるために造られたものだという事には全然気づかず、ただそれは停車場の構内を外国の遊戯場みたいに、複雑に楽しく、ハイカラにするためにのみ、設備せられてあるものだとばかり思っていました。しかも、かなり永い間そう思っていたのです。ブリッジの上ったり降りたりは、自分にはむしろ、ずいぶん垢抜けのした遊戯で、それは鉄道のサーヴィスの中でも、最も気のきいたサーヴィスの一つだと思っていたのですが、のちにそれはただ旅客が線路をまたぎ越えるための頗る実利的な階段に過ぎないのを発見して、にわかに興が覚めました。また、自分は子供の頃、絵本で地下鉄道というものを見て、これもやは

vision_old.txt - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

恥の多い生涯をっ1来ました。自分には、人間の生活というものカヾ、見
当っ力、ないのてす。自分は東北の田舎に生れましたのて、汽車をはじめ1見
たのは、よほ大きくなっ1力、らてした。自分はイ亭車場のフ・リッジを、上っ1、
降リ1、うし1れカヾ線足各をまたぎ越えるために造られたものだという事に
は全然気つ、、カ、す、ただれは停車場の構内を外国の遊戯場みたいに、複
雑に楽しく、ハイカラにするためにのみ、設備せられマあるものだとは・カ、リも
っ1いました。し力、も、カ、なリ水い間うっ1いたのてす。フ・リッジの上ったリ
降りたリは、自分にはむしろ、すいぶん垢抜けのした遊戯て、れは鉄道のサ
ーヴィスの中ても、最も気のきいたサーヴィスの一つだとっ1いたのてす力ヾ
の方にれはただ旅客ヵヾ線路をまたぎ越えるための頗る実利的な階段に過
ぎないのを癸見し1、にわ力、に興ヵヾ覚めました。また、自分は子供のヒ頁、絵
本て地下鉄道というものを見1、これもやは

Microsoft Cognitive Service OCRアクション

Vision API.txt - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

心の多い生涯を送って来ました。自分には、人間の生活というものが、見
当つかないのです。自分は東北の田舎に生れましたので、汽車をはじめて見
たのは、よほど大きくなってからでした。自分は停車場のブリッジを、上って、
降りて、そうしてそれが線路をまたぎ越えるために造られたものだという事に
は全然気づかず、ただそれは停車場の構内を外国の遊戯場みたいに、複
雑に楽しく、ハイカラにするためにのみ、設備せられてあるものだとはかり思
っていました。しかも、かなり永い間そう思っていたのです。ブリッジの上ったり
降りたりは、自分にはむしろ、ずいぶん垢抜けのした遊戯で、それは鉄道のサ
ーヴィスの中でも、最も気のきいたサーヴィスの一つだと思っていたのですが、
のちにそれはただ旅客が線路をまたぎ越えるための顔る実利的な階段に過
ぎないのを発見して、にわかに興が覚めました。また、自分は子供の頃、絵
本で地下鉄道というものを見て、これもやは

Google Cloud Visionサービステキスト検出アクション

C:¥Users¥ ¥Desktop¥Azure_Read_API.txt

恥の多い生涯を送って来ました。自分には、人間の生活というものが、見
当つかないのです。自分は東北の田舎に生れましたので、汽車をはじめて見
たのは、よほど大きくなってからでした。自分は停車場のブリッジを、上って、
降りて、そうしてそれが線路をまたぎ越えるために造られたものだという事に
は全然気づかず、ただそれは停車場の構内を外国の遊戯場みたいに、複
雑に楽しく、ハイカラにするためにのみ、設備せられてあるものだとばかり思
っていました。しかも、かなり永い間そう思っていたのです。ブリッジの上った
降りたりは、自分にはむしろ、ずいぶん垢抜けのした遊戯で、それは鉄道のサ
ーヴィスの中でも、最も気のきいたサーヴィスの一つだと思っていたのですが、
のちにそれはただ旅客が線路をまたぎ越えるための頬る実利的な階段に過
ぎないのを発見して、にわかに興が覚めました。また、自分は子供の頃、絵
本で地下鉄道というものを見て、これもやは

Azure Computer Vision v3.2

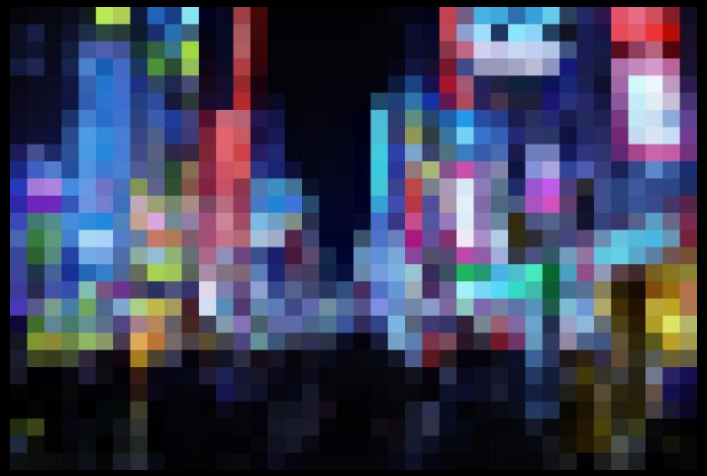
C:¥Users¥ ¥Desktop¥GCP_Document_AI.txt

恥の多い生涯を送って来ました。自分には、人間の生活というものが、見
当つかないのです。自分は東北の田舎に生れましたので、汽車をはじめて見
たのは、よほど大きくなってからでした。自分は停車場のブリッジを、上って、
降りて、そうしてそれが線路をまたぎ越えるために造られたものだという事に
は全然気づかず、ただそれは停車場の構内を外国の遊戯場みたいに、複
雑に楽しく、ハイカラにするためにのみ、設備せられてあるものだとばかり思
っていました。しかも、かなり永い間そう思っていたのです。ブリッジの上った
降りたりは、自分にはむしろ、ずいぶん垢抜けのした遊戯で、それは鉄道のサ
ーヴィスの中でも、最も気のきいたサーヴィスの一つだと思っていたのですが、
のちにそれはただ旅客が線路をまたぎ越えるための頗る実利的な階段に過
ぎないのを発見して、にわかに興が覚めました。また、自分は子供の頃、絵
本で地下鉄道というものを見て、これもやは

GCP Document AI

画像処理

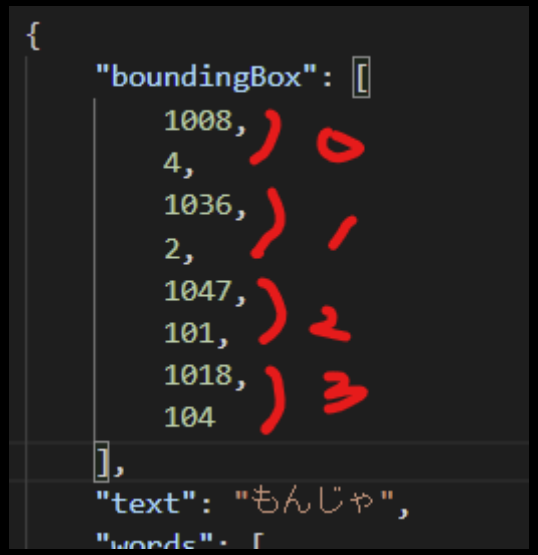
元画像、マスク画像、モザイク画像を合成



magick composite -compose src-over [モザイク画像] [元画像] [マスク画像] [出力画像]

画像処理

マスク画像生成




```
-draw "polygon 1008,4 1036,2 1047,101 1018,104"
74" -draw "polygon 53,473 118,473 118,483 53,483"
```

magick -size %OriginSize% xc:none -fill "#000000" %JoinedText% mask.png

画像処理

モザイク画像生成

幅x

-thumbnail 400

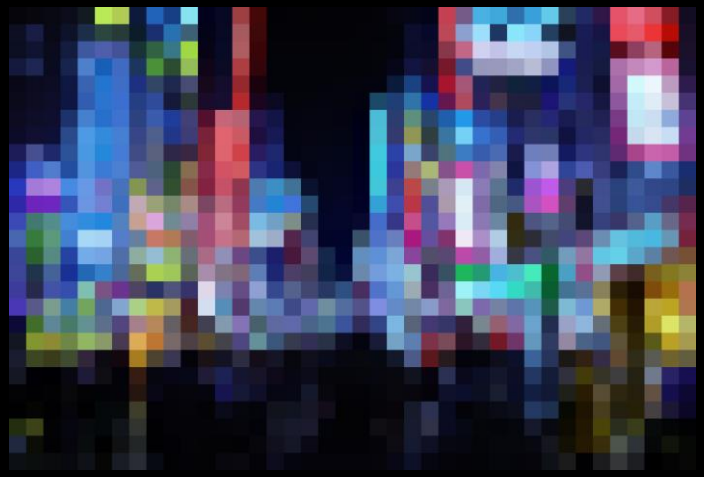
固定幅400に変更

-scale 10%

10%に縮小

-scale 1000*x/400 %

元の幅xに拡大



magick %SelectedFile% -thumbnail 400 -scale 10%% -scale %Bairitu%%% mosaic.png

まとめ

スクリプト実行アクション良いです！

Tips

- 2021年6月末現在、使用できるバージョンはIronPython 2.7.9
- 組込み関数以外の必要なモジュールはフォルダに入れておいて詳細から指定する
- PADからリストを渡したいときは文字列に変換してから
- Pythonスクリプトの実行アクションからの出力は文字列
- 文字列の末尾に余分な改行コードが含まれる
- 文字列の先頭にBOMが付加されている
- Webや数値としてはそのままでは使用できないので正規表現で抜き出す
- スクリプト側の%は%を重ねてエスケープする

Link

- IronPython.net https://ironpython.net/
- https://github.com/IronLanguages/ironpython2/releases/tag/ipy-2.7.11
- 組込み関数　https://docs.python.org/ja/2.7/library/functions.html
- モジュール　https://docs.python.org/ja/2.7/tutorial/modules.html
- Imagemagick https://imagemagick.org/index.php
- Azure Cognitive Service Computer vision v3.2 https://centraluseuap.dev.cognitive.microsoft.com/docs/services/computer-vision-v3-2/operations/5d986960601faab4bf452005

1 ブロック エラー発生時 BlockError

2 特別なフォルダーを取得
フォルダー デスクトップ のパスを取得し、SpecialFolderPath に保存する

3 コメント
##########################
APIキーの取得
##########################

4 特別なフォルダーを取得
フォルダー ドキュメント のパスを取得し、SpecialFolderPath2 に保存する

5 ファイルからテキストを読み取ります
ファイル SpecialFolderPath2 '\AzureOcrSecret.txt' の内容を読み取る

6 変数の設定
KEY を FileContents [0] に設定します

7 変数の設定
URL を FileContents [1] に設定します

8 コメント
##########################
ファイル選択とPOST
##########################

9 ファイルの選択ダイアログを表示
ファイルの選択ダイアログを表示し、選択したファイルを SelectedFile に保存する

10 If ButtonPressed = 'Cancel' then

11 フローを停止する フローを正常に停止します

12 End

13 テキストを置換する
SelectedFile 内でテキスト '\' を '/' に置き換え、結果を FilePath に保存する

14 Python スクリプトの実行
Python スクリプト 'body=open(' FilePath ', 'rb').read()

import httplib, urllib, base64

15 テキストの解析
テキスト PythonScriptOutput を位置 0 から解析して正規表現 'https?://[\w!?/+\-_~:,*&@#$'%'()'[\]]+' の最初の出現箇所のみ検索します。見つかった一致を Match に保存します

16 コメント
##########################
読み取り結果をGET
##########################

17 ラベル Result ラベル

18 Web サービスを呼び出します
Match のページにある Web サービスを呼び出して、応答を WebServiceResponse に保存します

19 JSON をカスタム オブジェクトに変換
JSON WebServiceResponse をカスタム オブジェクト JsonAsCustomObject に変換

20 {x} 変数の設定
Status を JsonAsCustomObject ['status'] に設定します

21 If Status <> 'succeeded' then

22 Wait 2 seconds

23 移動先 Result に移動する

24 End

25 コメント
##########################
座標データを描画コマンドに変換
##########################

26 新しいリストの作成
新しいリストを作成して List に保存

27 For each CurrentItem in JsonAsCustomObject ['analyzeResult']['readResults'][0]['lines']

28 項目をリストに追加
項目 '-draw "polygon ' CurrentItem ['boundingBox'][0],' CurrentItem ['boundingBox'][1] ' CurrentItem ['boundingBox'][2],' CurrentItem ['boundingBox'][3] ' CurrentItem ['boundingBox'][4],' CurrentItem ['boundingBox'][5] ' CurrentItem ['boundingBox'][6],' CurrentItem ['boundingBox'][7]'" をリスト List に追加

29 End

30 テキストの結合
リスト List の項目を で区切って結合

31 コメント
##########
画像処理開始フォルダ設定
##########

32 フォルダーの作成
フォルダー 'Demo' を SpecialFolderPath に作成

33 コメント
##########
マスク画像生成
##########

34 DOS コマンドの実行
DOS コマンドまたはコンソール アプリケーションを実行し、出力を OriginSize に保存する

35 DOS コマンドの実行
DOS コマンドまたはコンソール アプリケーションを実行する

36 コメント
##########
モザイク画像生成
##########

37 DOS コマンドの実行
DOS コマンドまたはコンソール アプリケーションを実行し、出力を WideSize に保存する

38 テキストを数値に変換
テキスト WideSize を数値に変換し、TextAsNumber に保存する

39 変数の設定
Bairitu を TextAsNumber * 2.5 に設定します

40 DOS コマンドの実行
DOS コマンドまたはコンソール アプリケーションを実行する

41 コメント
##########
画像の合成
##########

42 DOS コマンドの実行
DOS コマンドまたはコンソール アプリケーションを実行する

43 End